

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

PCT/JP2004/013391

08.09.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 3 年 1 1 月 1 2 日  
Date of Application:

出 願 番 号            特 願 2 0 0 3 - 3 8 2 5 7 0  
Application Number:

[ST. 10/C]:            [ J P 2 0 0 3 - 3 8 2 5 7 0 ]

出      願      人            松 下 電 器 産 業 株 式 会 社  
Applicant(s):

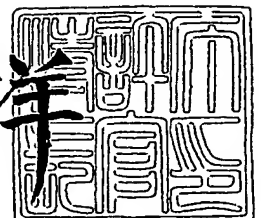


PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2 0 0 4 年 1 0 月 1 5 日

特許庁長官  
Commissioner,  
Japan Patent Office

小 川 洋



出証番号    出証特 2 0 0 4 - 3 0 9 2 9 6 5

【書類名】 特許願  
【整理番号】 2038250009  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 12/08 310  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 田中 哲也  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 岡林 はづき  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 中西 龍太  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 清原 督三  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 山本 崇夫  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 金子 圭介  
【特許出願人】  
    【識別番号】 000005821  
    【氏名又は名称】 松下電器産業株式会社  
【代理人】  
    【識別番号】 100109210  
    【弁理士】  
    【氏名又は名称】 新居 広守  
【手数料の表示】  
    【予納台帳番号】 049515  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 0213583

**【書類名】 特許請求の範囲****【請求項 1】**

N-ウェイ・セット・アソシエティブ方式のキャッシュメモリであって、  
N個のウェイのうち1つ以上のウェイを示す制御レジスタと、  
制御レジスタに示されるウェイをアクティブにする制御手段と、  
制御レジスタの内容を更新する更新手段と  
を備えることを特徴とするキャッシュメモリ。

**【請求項 2】**

前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイに対して少なくともリプレースを制限する

ことを特徴とする請求項 1 記載のキャッシュメモリ。

**【請求項 3】**

前記キャッシュメモリは、さらに、

ウェイ毎に設けられ、キャッシュデータのアドレスをタグとして保持するタグ保持手段と、

プロセッサから出力されるメモリアクセスアドレスの上位部分であるタグアドレスと、  
タグ保持手段から出力されるN個のタグとを比較することによりヒットかミスヒットかを  
判定するN個の比較手段を有し、

前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイに対応する  
比較手段をディスエーブルにする

ことを特徴とする請求項 1 記載のキャッシュメモリ。

**【請求項 4】**

前記制御手段は、さらに、制御レジスタに示されたアクティブなウェイ以外のウェイに  
対応するキャッシュアドレス保持手段に対して、比較手段へのタグ出力をディスエーブル  
にする

ことを特徴とする請求項 3 記載のキャッシュメモリ。

**【請求項 5】**

前記制御手段は、プロセッサからメモリアクセスアドレスが出力されたとき、当該アク  
セスアドレスについて、比較手段に最大2回のタグ比較を行わせるよう制御し、

1 回目のタグ比較において制御レジスタに示されたアクティブなウェイ以外のウェイに  
対応する比較手段をディスエーブルし、

1 回目のタグ比較においてミスヒットと判定された場合に、アクティブなウェイ以外の  
ウェイに対応する比較手段をディスエーブルしないで2 回目のタグ比較を行わせる

ことを特徴とする請求項 3 記載のキャッシュメモリ。

**【請求項 6】**

前記制御手段は、前記2 回目のタグ比較においてアクティブなウェイに対応する比較手  
段をディスエーブルする

ことを特徴とする請求項 5 記載のキャッシュメモリ。

**【請求項 7】**

前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイに対して、  
その状態の更新を禁止する

ことを特徴とする請求項 2 記載のキャッシュメモリ。

**【請求項 8】**

前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイについて、  
そのアクセス順序を示す情報の更新を禁止する

ことを特徴とする請求項 2 記載のキャッシュメモリ。

**【請求項 9】**

前記キャッシュメモリは、さらに

前記更新手段によって制御レジスタの内容が更新されたとき、ウェイに対するアクセス  
順序を示す情報をリセットするリセット手段を有する

ことを特徴とする請求項 2 記載のキャッシュメモリ。

【請求項 10】

前記更新手段は、

アクティブにすべきウェイを指定するウェイデータであって、タスク毎のウェイデータを保持する保持手段と、

実行中のタスクに対応するウェイデータを保持するよう前記制御レジスタを書き換える書き換え手段と

を有することを特徴とする請求項 2 記載のキャッシュメモリ。

【請求項 11】

前記保持手段は、メモリ中に記憶されたタスク毎のコンテキストデータの一部として前記ウェイデータを保持し、

前記書き換え手段は、タスク切り替えに際して、制御レジスタ中の現タスクのウェイデータをメモリに退避し、次タスクのウェイデータをメモリから前記制御レジスタに復帰する

ことを特徴とする請求項 10 記載のキャッシュメモリ。

【請求項 12】

前記保持手段は、タスク毎の前記ウェイデータを保持し、

前記書き換え手段は、

メモリに記憶された各タスクのアドレス範囲を記憶するアドレス記憶手段と、

アドレス記憶手段に記憶されたアドレス範囲と、プロセッサから出力される命令フェッチアドレスとに基づいて、実行中のタスクを判別する判別手段と、

判別された実行中のタスクに対応するウェイデータを前記保持手段から選択する選択手段と、

選択されたウェイデータを前記制御レジスタに書き込む書き込み手段と

を備えることを特徴とする請求項 11 記載のキャッシュメモリ。

【請求項 13】

前記保持手段は、タスク毎の前記ウェイデータを保持し、

前記書き換え手段は、

プロセッサから出力されるタスク番号に従って、実行中のタスクに対応するウェイデータを前記保持手段から選択する選択手段と、

選択されたウェイデータを前記制御レジスタに書き込む書き込み手段と

を備えることを特徴とする請求項 11 記載のキャッシュメモリ。

【請求項 14】

前記キャッシュメモリは、各ウェイにおけるリプレース単位をキャッシュエントリーのラインサイズと、ラインサイズの 2 の  $n$  乗分の 1 のサイズとに切り替え可能であり、

前記制御レジスタは、さらに、ウェイ毎のリプレースサイズを示し、

前記制御手段は、制御レジスタに示されたリプレースサイズを単位としてリプレース制御を行う

ことを特徴とする請求項 1 記載のキャッシュメモリ。

【請求項 15】

前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイに対して少なくともリプレースを制限し、制御レジスタに示されたアクティブなウェイに対して制御レジスタに示されたサイズを単位にリプレースを行う

ことを特徴とする請求項 14 記載のキャッシュメモリ。

【請求項 16】

前記更新手段は、

アクティブにすべきウェイを指定するウェイデータであってタスク毎のウェイデータと、タスク毎のリプレースサイズとを保持する保持手段と、

実行中のタスクに対応するウェイデータ及びリプレースサイズを保持するよう前記制御レジスタを書き換える書き換え手段と

を有することを特徴とする請求項 15 記載のキャッシュメモリ。

【請求項 17】

前記キャッシュメモリは、さらに、

キャッシュの単位となるデータを保持するキャッシュエントリー毎に、アクセスの有無を示す 1 ビットのアクセス情報を記憶する記憶手段と、

アクセス無しを示すアクセス情報に対応するキャッシュエントリーの中からリプレース対象のキャッシュエントリーを選択する選択手段と

を備えることを特徴とする請求項 1 記載のキャッシュメモリ。

【請求項 18】

N-ウェイ・セット・アソシエイティブ方式のキャッシュメモリを制御する制御方法であって、

N 個のウェイのうち 1 つ以上のウェイを示すウェイデータを制御レジスタに設定するステップと、

制御レジスタに示されるウェイをアクティブにする制御ステップと

を有することを特徴とする制御方法。

【請求項 19】

前記制御ステップでは、制御レジスタに示されたアクティブなウェイ以外のウェイに対して少なくともリプレースを制限する

ことを特徴とする請求項 18 記載の制御方法。

【請求項 20】

前記制御方法は、さらに、

アクティブにすべきウェイを指定するウェイデータであってタスク毎のウェイデータを保持する保持部から、実行中のタスクに対応するウェイデータ読み出して、読み出したウェイデータを前記制御レジスタに書き込む更新ステップを有する

ことを特徴とする請求項 19 記載の制御方法。

【書類名】明細書

【発明の名称】 キャッシュメモリおよびその制御方法

【技術分野】

【0001】

本発明は、プロセッサのメモリアクセスを高速化するためのキャッシュメモリおよびその制御方法に関する。

【背景技術】

【0002】

キャッシュメモリは、主記憶装置のアクセス時間を短縮しプロセッサの処理能力の向上を図るために、従来から広く用いられている。

特許文献1等の開示出されたキャッシュメモリは、メインメモリのブロック単位のデータを各エントリに格納し、このエントリを介して、マルチタスク処理を行なう処理ユニットからのアクセスに対応するデータの転送制御と排他制御を行なう。このキャッシュメモリは、エントリに格納されているブロックを排他制御の対象として設定した処理ユニットの各タスクの識別情報を登録するタスク識別情報登録部を、エントリごとに設ける構成とし、タスク単位で、エントリに格納されているブロックの排他制御、および、この排他制御の設定と解除を行なっている。

このキャッシュメモリによれば、マルチタスク処理における排他制御を効率良く行ない、タスク間で共通に使用するデータの矛盾を解消することを図っている。

【特許文献1】特開平6-266620号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

しかしながら、上記従来技術におけるキャッシュメモリによれば、プロセッサのタスク切り替えに伴ってキャッシュメモリのヒット率が実行中でない他のタスクによる影響を受けるといった問題がある。

例えば、タスクAの命令列（又はデータ）がキャッシュメモリに格納されている状態でタスクAの実行からタスクBの実行に切り換えられた場合、タスクBの実行によりキャッシュメモリ中のタスクAの命令列（又はデータ）が追い出されてしまう。タスクAの命令列（又はデータ）がキャッシュメモリから追い出されていれば、再度タスクAが実行されたときに、キャッシュミスが発生するという問題がある。特に、圧縮音声データや圧縮映像データのデコード／エンコード処理などのリアルタイム性を必要とする処理では、上記タスク切り替えに伴う他のタスクの影響によって、タスク切り替え後のキャッシュのリブレース処理によってタスクの割当時間を侵食され、必要な処理時間を確保できず、リアルタイム性が損なわれるあるいは処理時間を確定できないという問題がある。

【0004】

本発明は、タスク切り替え等によるキャッシュメモリの他のタスクの影響を防止し、タスクの実質的な処理時間を容易に確保するキャッシュメモリを提供することを目的とする。

【課題を解決するための手段】

【0005】

上記課題を解決するため本発明のキャッシュメモリは、N-ウェイ・セット・アソシエイティブ方式のキャッシュメモリであって、N個のウェイのうち1つ以上のウェイを示す制御レジスタと、制御レジスタに示されるウェイをアクティブにする制御手段と、制御レジスタの内容を更新する更新手段とを備える。

この構成によれば、キャッシュメモリを構成するN-ウェイのうち、制御レジスタに示されたウェイのみをアクティブにし、しかも制御レジスタの内容は更新可能なので、プロセッサが実行する処理に応じてアクティブなウェイを動的に設定することができる。タスクとウェイとを対応付ければ、タスク切り替え後に他のタスクによりキャッシュメモリから必要なデータが追い出されることが解消され、タスク切り替えに伴うヒット率の他のタ

スクからの影響を防止することができる。その結果、タスクに必要とされる実質的な処理時間を容易に確保することができる。

**【0006】**

ここで、前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイ（インアクティブなウェイと呼ぶ）に対して少なくともリプレースを制限する構成としてもよい。

ここで、前記キャッシュメモリは、さらに、ウェイ毎に設けられ、キャッシュデータのアドレスをタグとして保持するタグ保持手段と、プロセッサから出力されるメモリアクセスアドレスの上位部分であるタグアドレスと、タグ保持手段から出力されるN個のタグとを比較することによりヒットかミスヒットかを判定するN個の比較手段を有し、前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイに対応する比較手段をディスエーブルにする構成としてもよい。

**【0007】**

この構成によれば、インアクティブなウェイに対応する比較手段をディスエーブルにするので、比較手段における消費電力を低減することができる。

ここで、前記制御手段は、さらに、制御レジスタに示されたアクティブなウェイ以外のウェイに対応するタグ保持手段に対して、比較手段へのタグ出力をディスエーブルにする構成としてもよい。

**【0008】**

この構成によれば、インアクティブなウェイに対応するタグ出力と比較手段とがディスエーブルされるので、タグ保持手段の消費電力を低減することができる。

ここで、前記制御手段は、プロセッサからメモリアクセスアドレスが出力されたとき、当該アクセスアドレスについて、比較手段に最大2回のタグ比較を行わせるよう制御し、1回目のタグ比較では、制御レジスタに示されたアクティブなウェイ以外のウェイに対応する比較手段をディスエーブルし、1回目のタグ比較においてミスヒットと判定された場合に、アクティブなウェイ以外のウェイに対応する比較手段をディスエーブルしないで2回目のタグ比較を行わせる構成としてもよい。

**【0009】**

この構成によれば、1回目のタグ比較におけるヒット率が高いほど、比較手段における消費電力を低減することができ、しかも、1回目のタグ比較においてミスヒットした場合に2回目のタグ比較を行うので、インアクティブなウェイのキャッシュデータも有効に活用することができる。

**【0010】**

ここで、前記制御手段は、前記2回目のタグ比較においてアクティブなウェイに対応する比較手段をディスエーブルする構成としてもよい。

この構成によれば、2回目のタグ比較ではインアクティブなウェイに対応する比較手段のみがタグ比較を行うので、さらに消費電力を低減することができる。

ここで、前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイに対して、その状態の更新を禁止する構成としてもよい。

この構成によれば、例えばインアクティブなウェイの状態を示すフラグ類の更新を禁止することにより、インアクティブなウェイに対するタスク切り替えによる影響を防止することができる。

**【0011】**

ここで、前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイについて、そのアクセス順序を示す情報の更新を禁止する構成としてもよい。

この構成によれば、アクセス順序を示す情報の更新を禁止するので、更新手段による更新によってインアクティブなウェイがアクティブなウェイに切り替わった直後に、リプレースされてしまう可能性を小さくする。

**【0012】**

ここで、前記キャッシュメモリは、さらに、前記更新手段によって制御レジスタの内容

が更新されたとき、ウェイに対するアクセス順序を示す情報をリセットするリセット手段を有する構成としてもよい。

この構成によれば、インアクティブなウェイがインアクティブであるがゆえにアクセス順序が古くなった場合に、インアクティブなウェイがアクティブなウェイに切り替わった直後にリプレースされてしまう可能性を小さくする。

#### 【0013】

この構成によれば、インアクティブなウェイについては、少なくともリプレースが制限される。つまり、インアクティブなウェイについて完全にディスエーブルにしても、リプレースだけをイネーブルにしてもよい。後者の場合、キャッシュメモリに対するリード/ライトまでは制限されないので、ヒット率の低化を防止し、かつインアクティブなウェイを有効に活用することができる。

#### 【0014】

ここで、前記更新手段は、アクティブにすべきウェイを指定するウェイデータであって、タスク毎のウェイデータを保持する保持手段と、実行中のタスクに対応するウェイデータを保持するよう前記制御レジスタを書き換える書き換え手段とを有する構成としてもよい。

この構成によれば、タスクが切り替わる毎に動的に、制御レジスタを書き換えるので、他タスク毎にアクティブなウェイを対応付けることができる。

#### 【0015】

ここで、前記保持手段は、メモリ中に記憶されたタスク毎のコンテキストデータの一部として前記ウェイデータを保持し、前記書き換え手段は、タスク切り替えに際して、制御レジスタ中の現タスクのウェイデータをメモリに退避し、次タスクのウェイデータをメモリから前記制御レジスタに復帰する構成としてもよい。

この構成によれば、制御レジスタの更新は、OS (Operating System) によるタスク切り替えにより、キャッシュメモリのハードウェアを大幅に追加することなく簡単に実現することができる。

#### 【0016】

ここで、前記保持手段は、タスク毎の前記ウェイデータを保持し、前記書き換え手段は、メモリに記憶された各タスクのアドレス範囲を記憶するアドレス記憶手段と、アドレス記憶手段に記憶されたアドレス範囲と、プロセッサから出力される命令フェッチアドレスとに基づいて、実行中のタスクを判別する判別手段と、判別された実行中のタスクに対応するウェイデータを前記保持手段から選択する選択手段と、選択されたウェイデータを前記制御レジスタに書き込む書き込み手段とを備える構成としてもよい。

#### 【0017】

この構成によれば、制御レジスタの更新は、キャッシュメモリ自身が主体的に判断することにより行われるので、どのようなプロセッサに対しても、タスク毎に対応するウェイをアクティブにすることができる。

#### 【0018】

ここで、前記保持手段は、タスク毎の前記ウェイデータを保持し、前記書き換え手段は、プロセッサから出力されるタスク番号に従って、実行中のタスクに対応するウェイデータを前記保持手段から選択する選択手段と、選択されたウェイデータを前記制御レジスタに書き込む書き込み手段とを備える構成としてもよい。

この構成によれば、プロセッサから出力されるタスク番号を利用するのでハードウェアを大幅に追加することなく、制御レジスタを簡単に更新し、タスク毎に対応するウェイをアクティブにすることができる。

ここで、前記キャッシュメモリは、各ウェイにおけるリプレース単位をキャッシュエントリーのラインサイズと、ラインサイズの2の $n$ 乗分の1のサイズとに切り替え可能であり、前記制御レジスタは、さらに、ウェイ毎のリプレースサイズを示し、前記制御手段は、制御レジスタに示されたリプレースサイズを単位としてリプレース制御を行う構成としてもよい。



## 【0019】

また、前記制御手段は、制御レジスタに示されたアクティブなウェイ以外のウェイに対して少なくともリプレースを制限し、制御レジスタに示されたアクティブなウェイに対して制御レジスタに示されたサイズを単位にリプレースを行う構成としてもよい。

ここで、前記更新手段は、アクティブにすべきウェイを指定するウェイデータであってタスク毎のウェイデータと、タスク毎のリプレースサイズとを保持する保持手段と、実行中のタスクに対応するウェイデータ及びリプレースサイズを保持するよう前記制御レジスタを書き換える書き換え手段とを有する構成としてもよい。

## 【0020】

この構成によれば、タスク毎にアクティブなウェイを切り換えると同時に、リプレース単位をも切り換えることができるので、タスクの処理内容に応じてミスヒットを低減することができる。

また、本発明のキャッシュメモリの制御方法は、N-ウェイ・セット・アソシエイティブ方式のキャッシュメモリを制御する制御方法であって、N個のウェイのうち1つ以上のウェイを示すウェイデータを制御レジスタに設定するステップと、制御レジスタに示されるウェイをアクティブにする制御ステップとを有する。

## 【0021】

ここで、前記制御ステップでは、制御レジスタに示されたアクティブなウェイ以外のウェイに対して少なくともリプレースを制限するようにしてもよい。

ここで、前記制御方法は、さらに、アクティブにすべきウェイを指定するウェイデータであってタスク毎のウェイデータを保持する保持部から、実行中のタスクに対応するウェイデータ読み出して、読み出したウェイデータを前記制御レジスタに書き込む更新ステップを有する構成としてもよい。

## 【発明の効果】

## 【0022】

本発明のキャッシュメモリによれば、プロセッサが実行する処理毎にアクティブなウェイを動的に設定することができるので、タスクとウェイとを対応付ければ、タスク切り替え後に他のタスクによりキャッシュメモリから必要なデータが追い出ることが解消され、タスク切り替えに伴うヒット率の他のタスクからの影響を防止することができる。その結果、タスクに必要とされる実質的な処理時間を容易に確保することができる。

## 【発明を実施するための最良の形態】

## 【0023】

## (実施の形態1)

図1は、本発明の実施の形態1におけるプロセッサ1、キャッシュメモリ3、メモリ2を含むシステムの概略構成を示すブロック図である。同図のように、本発明のキャッシュメモリ3は、プロセッサ1およびメモリ2を有するシステムに備えられる。プロセッサ1は、マルチタスク制御を行うプロセッサであり、メモリ2中のタスク1～4等を切り替えて実行する。キャッシュメモリ3は、N-ウェイ・セット・アソシエイティブ方式のキャッシュメモリであって、タスク毎にN個のウェイのうち1つ以上のウェイを対応させて、当該タスク実行中に対応するウェイをアクティブにするよう構成されている。各タスクは、アクティブなウェイに対しては、キャッシュメモリとしての全機能を利用可能である。また、各タスクは、アクティブでないウェイ（インアクティブなウェイと呼ぶ）に対しては、キャッシュメモリとしての全機能のうち利用可能な機能が制限されている。本実施の形態では、インアクティブなウェイは、リプレースする機能が制限され、それ以外のリード、ライト等は制限されていないものとする。

## 【0024】

以下では、キャッシュメモリ3の具体例として、8ウェイ・セット・アソシエイティブ方式のキャッシュメモリに本発明を適用した場合の構成について説明する。

図2は、キャッシュメモリ3の構成例を示すブロック図である。同図のように、キャッシュメモリ3は、アドレスレジスタ20、デコーダ30、8つのウェイ31a～31h（

以下ウェイ 0～7 と略す)、8つの比較器 32a～32h、8つのアンド回路 33a～33h、オア回路 34、セクタ 35、セクタ 36、デマルチプレクサ 37、制御部 38 を備える。

#### 【0025】

アドレスレジスタ 20 は、メモリ 2 へのアクセスアドレスを保持するレジスタである。このアクセスアドレスは 32 ビットであるものとする。同図に示すように、アクセスアドレスは、最上位ビットから順に、21 ビットのタグアドレス、4 ビットのセットインデックス (図中の SI)、5 ビットのワードインデックス (図中の WI) を含む。

ここで、タグアドレスはウェイにマッピングされるメモリ中の領域 (そのサイズはセット数×ブロックである。) を指す。この領域のサイズは、タグアドレスよりも下位のアドレスビット (A10～A0) で定まるサイズつまり 2k バイトであり、1つのウェイのサイズでもある。セットインデックス (SI) はウェイ 0～3 に跨る複数セットの 1つを指す。このセット数は、セットインデックスが 4 ビットなので 16 セットである。タグアドレスおよびセットインデックスで特定されるブロックは、リプレース単位であり、キャッシュメモリに格納されている場合はラインデータ又はラインと呼ばれる。ラインデータのサイズは、セットインデックスよりも下位のアドレスビットで定まるサイズつまり 128 バイトである。1ワードを 4 バイトとすると、1ラインデータは 32 ワードである。ワードインデックス (WI) は、ラインデータを構成する複数ワード中の 1ワードを指す。アドレスレジスタ 20 中の最下位 2 ビット (A1、A0) は、ワードアクセス時には無視される。

#### 【0026】

デコーダ 30 は、セットインデックスの 4 ビットをデコードし、8つのウェイ 0～7 の同順に位置するキャッシュエントリからなる 16 個のセット中の 1つのセットを選択する。

ウェイ 0～7 は、同じ構成を有数する 8つのウェイであり、8×2k バイトの容量を有する。ウェイ 0 は、16 個のキャッシュエントリを有する。1つのキャッシュエントリは、バリッドフラグ V、21 ビットのタグ、128 バイトのラインデータを保持する。バリッドフラグ V は、そのキャッシュエントリが有効か否かを示す。タグは 21 ビットのタグアドレスのコピーである。ラインデータは、タグアドレスおよびセットインデックスにより特定されるブロック中の 128 バイトデータのコピーである。また、ウェイ 1～7 についても、ウェイ 0 と同様である。セットインデックスの 4 ビットによってデコーダ 30 を介して選択される 4 ウェイに跨る 4つのキャッシュエントリは、セットと呼ばれる。また、同図では書き込みがあったことを示すダーティフラグは省略されている。

#### 【0027】

比較器 32a は、アドレスレジスタ 20 中のタグアドレスと、セットインデックスにより選択されたセットに含まれる 4つのタグ中のウェイ 0 のタグとが一致するか否かを比較する。比較器 32b～32h についても、ウェイ 31b～31h に対応すること以外は同様である。

アンド回路 33a は、バリッドフラグと比較器 32a の比較結果とが一致するか否かを比較する。この比較結果を h0 とする。比較結果 h0 が 1 である場合は、アドレスレジスタ 20 中のタグアドレスおよびセットインデックスに対応するラインデータが存在すること、つまりウェイ 0 においてヒットしたことを意味する。比較結果 h0 が 0 である場合は、ミスヒットしたことを意味する。アンド回路 33b～33h についても、ウェイ 31b～31h に対応すること以外は同様である。その比較結果 h1～h7 は、ウェイ 1～7 でヒットしたかミスしたかを意味する。

#### 【0028】

オア回路 34 は、比較結果 h0～h3 のオアをとる。このオアの結果を hit とする。hit は、キャッシュメモリにヒットしたか否かを示す。

セクタ 35 は、選択されたセットにおけるウェイ 0～7 のラインデータのうち、ヒットしたウェイのラインデータを選択する。

セレクト36は、セレクト35により選択された32ワードのラインデータのうち、ワードインデックスに示される1ワードを選択する。

#### 【0029】

デマルチプレクサ37は、キャッシュエントリにデータを書き込む際に、ウェイ0～7の1つに書き込みデータを出力する。この書き込みデータはワード単位でよい。

制御部38は、内部にウェイ・レジスタ371を有し、キャッシュメモリ3の全体の制御を行う。ウェイレジスタ371は、ウェイ0～7のうちアクティブなウェイを示すデータを保持するレジスタである。制御部38は、ウェイ・レジスタ371によって示されるアクティブなウェイに対しては、キャッシュメモリとしての全機能の制限なく制御し、インアクティブなウェイに対しては、リプレースする機能を制限する。

#### 【0030】

図3は、ウェイ・レジスタ371のビット構成を示す図である。同図のように、ウェイ・レジスタ371は、32ビットレジスタであり、下位8ビットにウェイ0～7に対応するW0フラグ～W7フラグを保持する。例えば、W0フラグが1のときウェイ0がアクティブなウェイであることを示し、0のときウェイ0がインアクティブなウェイであることを示す。W1フラグ～W7フラグについても同様である。以下、W0フラグ～W7フラグの集まりをアクティブウェイデータと呼ぶ。このウェイ・レジスタ371は、プロセッサ1から直接読み書き可能であり、各タスクのコンテキストの一部をなす。つまり、タスク毎にアクティブウェイデータを有し、タスク切り替えによって、ウェイ・レジスタ371の内容は実行中のタスクに対応するアクティブウェイデータに書き換えられる。

#### 【0031】

図4は、ウェイ・レジスタ371とウェイとの対応関係を示す説明図である。同図左側では、ウェイ・レジスタ371に保持されているアクティブウェイデータが“00111000”であるので、ウェイ2、3、4がアクティブウェイとなり、ウェイ0、1、5、6、7がインアクティブになる。タスク切り替えに際して、ウェイ・レジスタ371は、例えば同図右側のようなアクティブウェイデータに書き換えられる。同図右側では、ウェイ5～7がアクティブとなり、ウェイ0～4がインアクティブになる。

#### 【0032】

図5は、制御部38におけるリプレース処理を示すフローチャートである。同図において、制御部38は、ミスヒットが発生したか否かを判定し(S51)、ミスヒットが発生したと判定された場合に、セットインデックスにより選択されたセットにおける、4つウェイのキャッシュエントリの中からリプレース対象を1つ選択する(ステップS52)。このリプレース対象の選択はLRU方式でよい。

#### 【0033】

さらに、制御部38は、ウェイ・レジスタ371を参照して、選択されたウェイがアクティブであるか否かを判定し(S53)、アクティブでなければステップS52に戻り再度他のウェイのキャッシュエントリを選択する。制御部38は選択されたアクティブなウェイのキャッシュエントリをリプレースする(S54)。

このように、制御部38は、ウェイ・レジスタ371に示されるインアクティブなウェイに対しては、リプレースを制限し、制御部38は、アクティブなウェイに対しては、リプレースを制限することなくキャッシュメモリとしても全機能を制御する。ここでは、リプレースの制限はリプレースの禁止としている。

#### 【0034】

図6は、プロセッサ1におけるタスク切り替え処理を示すフローチャートである。タスク切り替え処理は時間の経過やイベントの発生により起動される。同図においてプロセッサ1は、現在実行中のタスクのコンテキストをメモリ2中の例えばスタック領域に退避させる(ステップS61)、次に実行すべきタスクのコンテキストをスタック領域から復帰させる(ステップS62)。ここで、スタック領域は、図7に示すように、メモリ2に確保され、各タスクのコンテキストを記憶するための領域である。各タスクのコンテキストは、プロセッサの汎用レジスタのデータや、種々の制御レジスタのデータを含み、加えて、本

実施の形態ではウェイレジスタに格納されるアクティブウェイデータを含む。

#### 【0035】

このようにして、ウェイ・レジスタ371は、タスク切り替えに際して書き換えられるので、常に実行中のタスクに対応するアクティブウェイデータを保持することになる。

以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、プロセッサ1に実行されるタスクから見れば、キャッシュメモリへのリードおよびライトについてはアクティブなウェイに対してもインアクティブなウェイに対しても可能であるが、ミスヒットした場合にリプレース対象となるウェイについてはアクティブなウェイに制限されることになる。

#### 【0036】

例えば、図4において同図左側をタスク1実行時、右側をタスク2実行時のアクティブウェイとする。タスクの実行が経過するにつれて、タスク1のキャッシュデータは次第にウェイ2～3に格納されていき、タスク2のキャッシュデータは次第にウェイ4～7に格納されていくことになる。言い換えれば、ウェイ2～3に格納されたタスク1のキャッシュデータはタスク2の実行によって追い出されない（リプレースされない）。また、ウェイ4～7に格納されたタスク2のキャッシュデータはタスク1の実行によって追い出されない（リプレースされない）。その結果、タスク切り替えに伴ってタスク1では必要なキャッシュデータが、他のタスクによってリプレースされ、再度タスク1実行時に追い出されたデータをキャッシュにリプレースすることも解消できる。その結果、タスク切り替えに伴う無駄なリプレースの発生を防止させることができ、他のタスクからの影響を抑えることができる。

#### 【0037】

##### <変形例>

なお、本発明のキャッシュメモリは、上記の実施形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。

(1) 複数のタスクと複数のウェイの対応関係については、1つのタスクに独占されるウェイと、複数のタスクに共用されるウェイとを混在させることができる。例えば、図4において、ウェイ2～4はタスク1が独占し、ウェイ5～7はタスク2が独占し、ウェイ0、1は他のタスクが共用するものとする。この場合、タスク1および2は、ウェイを独占するので、タスク切り替えによるキャッシュミス低減し、リアルタイム性を要する処理に適している。タスク1および2以外のタスクは、リアルタイム性を要しないイベントドリブンな処理等に適している。

#### 【0038】

(2) 上記実施の形態では、制御部38は、アクティブなウェイについてはキャッシュメモリの全機能を制御し、インアクティブなウェイについてはリプレースを禁止しているが、これに限らない。

例えば、制御部38は、ウェイ・レジスタ371に示されたインアクティブなウェイに対して、その状態の更新を禁止する構成としてもよい。例えば、ウェイの状態を表すフラグ類の更新を禁止することにより、インアクティブなウェイに対するタスク切り替えによる影響を防止することができる。

また、制御部38は、ウェイ・レジスタ371に示されたインアクティブなウェイについて、そのアクセス順序を示す情報の更新を禁止する構成としてもよい。これによれば、アクセス順序を示す情報の更新を禁止するので、更新手段による更新によってインアクティブなウェイがアクティブなウェイに切り替わった直後に、リプレースされてしまう可能性を小さくする。

#### 【0039】

あるいは、制御部38は、インアクティブなウェイについて、全機能を禁止するようにしてもよい。この場合、インアクティブなウェイのタグ出力を禁止するよう出力イネーブル信号をディスエーブルにすればよい。こうすれば、インアクティブなウェイの消費電力を低減することができる。また、全機能を禁止する場合には、各タスクがウェイを共用す

ることなく独占するように、タスクとウェイとを対応付けることが望ましい。こうすれば、メモリとキャッシュメモリとの間でデータに矛盾が生じることを防止することができる。

また、制御部 38 は、リプレースの禁止に加えて、アクセスの順番を示す LRU 用の順序データを更新しないように構成してもよい。

#### 【0040】

(3) ウェイ・レジスタ 371 の内容が更新されたとき、LRU 方式で用いられるアクセス順序情報をリセットする構成としてもよい。こうすれば、インアクティブなウェイがインアクティブであるがゆえにアクセス順序が古くなった場合に、インアクティブなウェイがアクティブなウェイに切り替わった直後にリプレースされてしまう可能性を小さくすることができる。

#### 【0041】

(4) また、制御部 38 は、リプレース禁止の代わりにリプレース回数を制限する構成としてもよいし、ウェイ中の特定のキャッシュエントリーに対するリプレースを禁止し、その他のキャッシュエントリーに対してはリプレースを行う構成としてもよい。

#### 【0042】

(5) 上記実施の形態では、8 ウェイ・セット・アソシエイティブのキャッシュメモリを例に説明したが、ウェイ数は、4 ウェイでも 16 ウェイでもいくつでもよい。また、上記実施の形態では、セット数が 16 である例を説明したが、セット数はいくつでもよい。

#### 【0043】

(6) 上記実施の形態では、セット・アソシエイティブのキャッシュメモリを例に説明したが、フル・アソシエイティブ方式のキャッシュメモリであってもよい。フル・アソシエイティブ方式の場合、セットが 1 つのケースと考えることができる。

#### 【0044】

##### (実施の形態 2)

実施の形態 1 では、ウェイレジスタ 371 をタスク切り替えによって書き換える構成を説明したが、本実施の形態では、キャッシュメモリにおいてタスクを判別して判別結果に応じてウェイレジスタ 371 を書き換える構成について説明する。加えて、実施の形態 1 ではリプレースアルゴリズムが周知の LRU 方式としたが、本実施の形態ではアクセス順序を示すデータの代わりに 1 ビットのフラグを用いる擬似的な LRU 方式を行う構成について説明する。

#### 【0045】

図 8 は、本発明の実施の形態 2 におけるキャッシュメモリの構成を示すブロック図である。同図のキャッシュメモリは、図 2 の構成と比較して、ウェイ 31a~31d の代わりにウェイ 131a~131d を備える点と、制御部 38 の代わりに制御部 138 を備える点とが異なっている。以下、同じ点は説明を省略して、異なる点を中心に説明する。

ウェイ 131a は、ウェイ 31a と比べて、各キャッシュエントリー中に、使用フラグとニューフラグとが追加されている点が異なる。図 9 に、キャッシュエントリーのビット構成を示す。1 つのキャッシュエントリーは、バリッドフラグ V、21 ビットのタグ、128 バイトのラインデータ、使用フラグ U、ニューフラグ N およびダーティフラグ D を保持する。このうち、使用フラグ U は、そのキャッシュエントリーにアクセスがあったか否かを示し、ミスヒットによるリプレースに際してセット内の 8 つのキャッシュエントリーにおけるアクセス順序の代わりに用いられる。より正確には、使用フラグ U の 1 は、アクセスがあったことを、0 はないことを意味する。セット内の 8 つの使用フラグは、全て 1 になれば、0 にリセットされるので、セット内の 8 つのキャッシュエントリーにおける使用の有無を示す相対的な値である。別言すれば、使用フラグ U は、アクセスされた時期が古いか新しいか 2 つの相対的な状態を示す。つまり、使用フラグ U が 1 のキャッシュエントリーは、使用フラグが 0 のキャッシュエントリーよりも新しくアクセスされたことを意味する。また、ニューフラグ N は、リプレース直後（又はフィル直後）に初期値として 1 が設定され、当該キャッシュエントリーへのアクセスがあったときに 0 にリセットされる。

。つまり、ニューフラグNの1は、当該キャッシュエントリーがリプレイス（又はフィル）されてから一度もアクセスされていない、新しい状態であることを意味する。

#### 【0046】

制御部138は、制御部38と比べて、設定部372が追加された点と、使用フラグUおよびニューフラグNの設定および更新を行う点とが異なる。

設定部372は、プロセッサ1において実行されているタスクを判別し、判別したタスクに対応するアクティブウェイデータをウェイ・レジスタ371に設定する。

#### 【0047】

##### <設定部の構成>

図10は、設定部372の構成例を示すブロック図である。同図のように、設定部372は、判別部100a~100dとアクティブウェイデータ保持部110a~110dとセレクト111とを備える。

#### 【0048】

判別部100aは、スタートアドレス保持部101、エンドアドレス保持部102、比較器103、104、アンド回路105とを有し、実行中のタスクがタスク1であるかを判別する。

スタートアドレス保持部101、エンドアドレス保持部102は、プロセッサ1から読み書き可能であり、メモリ2に格納されたタスク1のスタートアドレス、エンドアドレスをそれぞれ保持する。このスタートアドレスおよびエンドアドレスは、プロセッサ1によって予め書き込まれ、動的に変更可能である。

#### 【0049】

比較器103は、プロセッサ1から出力される命令フェッチアドレス（IFアドレス）とスタートアドレス保持部101から出力されるスタートアドレスとを比較し、スタートアドレスよりもIFアドレスの方が大きい場合に1を出力する。

比較器104は、プロセッサ1から出力されるIFアドレスとエンドアドレス保持部102から出力されるエンドアドレスとを比較し、IFアドレスよりもエンドアドレスの方が大きい場合に1を出力する。

#### 【0050】

アンド回路105は、比較器103および104の比較結果が共に1の場合、すなわち、IFアドレスがタスク1の命令をフェッチしている場合に、実行されているタスクがタスク1であることを示す。

判別部100b~100dについても同様であり、実行中のタスクがタスク2~3であるかを判別する。

#### 【0051】

アクティブウェイデータ保持部110a~110dは、プロセッサ1から読み書き可能であり、判別部100a~100dに対応するタスクのアクティブウェイデータを保持する。このアクティブウェイデータは、プロセッサ1によって予め書き込まれ、動的に変更可能である。

セレクト111は、判別部100a~100dの判別結果に従って、実行中のタスクに対応するアクティブウェイデータを選択し、ウェイ・レジスタ371に出力する。これにより、ウェイ・レジスタ371は、実行中のタスクに対応するアクティブウェイデータを保持する。

#### 【0052】

##### <使用フラグの更新例>

図11は、制御部138による使用フラグUの更新例を示す説明図である。同図では、説明の便宜上8ウェイではなく4ウェイの場合について説明する。同図の上段、中断、下段は、ウェイ0~3に跨るセットNを構成する4つのキャッシュエントリーを示している。4つのキャッシュエントリー右端の1又は0は、それぞれ使用フラグの値である。この4つの使用フラグUをU0~U3と記す。

#### 【0053】

同図上段では(U0~U3)=(1, 0, 1, 0)であるので、ウェイ0、2のキャッシュエントリはアクセスがあったことを、ウェイ1、3のキャッシュエントリはアクセスがないことを意味する。

この状態で、メモリアクセスがセットN内のウェイ1のキャッシュエントリにヒットした場合、同図中断に示すように、(U0~U3)=(1, 1, 1, 0)に更新される。つまり、実線に示すようにウェイ1の使用フラグU1が0から1に更新される。

#### 【0054】

さらに、同図中断の状態で、メモリアクセスがセットN内のウェイ3のキャッシュエントリにヒットした場合、同図下断に示すように、(U0~U3)=(0, 0, 0, 1)に更新される。つまり、実線に示すようにウェイ3の使用フラグU1が0から1に更新される。加えて、破線に示すようにウェイ3以外の使用フラグU0~U2が1から0に更新される。これにより、ウェイ3のキャッシュエントリが、ウェイ0~2の各キャッシュエントリよりも新しくアクセスされたことを意味することになる。

#### 【0055】

制御部138は、キャッシュミス時に使用フラグに基づいてリプレース対象のキャッシュエントリを決定してリプレースを行う。例えば、制御部138は、図11上段では、ウェイ1とウェイ3の何れかをリプレース対象と決定し、図11中断ではウェイ3をリプレース対象と決定し、図11下段ではウェイ0~2の何れかをリプレース対象と決定する。

#### 【0056】

<使用フラグ、ニューフラグの更新処理>

図12は、制御部138における使用フラグおよびニューフラグのフラグ更新処理を示すフローチャートである。同図では、バリッドフラグが0(無効)であるキャッシュエントリの使用フラグUは0に初期化されているものとする。

#### 【0057】

同図において、制御部138は、キャッシュヒットしたとき(ステップS61)、セットインデックスにより選択されたセットにおけるヒットしたウェイの使用フラグUを1にセットし(ステップS62)、選択されたセット内のヒットしたウェイのキャッシュエントリのニューフラグが1なら0にリセットする(ステップS171)。

さらに、制御部138は、そのセット内の他のウェイの使用フラグUを読み出し(ステップS63)、読み出した使用フラグUが全て1であるか否かを判定し(ステップS64)、全て1でなければ終了し、全て1であれば他のウェイの全ての使用フラグUを0にリセットする(ステップS65)。

#### 【0058】

このようにして制御部138は、図11に示した更新例のように、使用フラグを更新する。また、ニューフラグNは、キャッシュエントリのリプレース後、最初にアクセスされた時点でリセットされる。

#### 【0059】

<リプレース処理>

図13は、制御部138におけるリプレース処理フローを示す図である。同図において制御部138は、メモリアクセスがミスしたとき(ステップS91)、セットインデックスにより選択されたセットにおける、8つウェイの使用フラグUと、8つのニューフラグN0~N7を読み出し(ステップS92)、読み出した8つのニューフラグN0~N7の全てが1であるか否かを判定し(ステップS161)、全てが1である場合は、ステップS93に進み、全てが1ではない(0がある)場合には、使用フラグUが0のウェイのうち、ニューフラグNが1のウェイを除外する(ステップS162)。

#### 【0060】

さらに、制御部138は、使用フラグUが0のウェイを1つ選択する(ステップS93)。このとき、使用フラグUが0になっているウェイが複数存在する場合は、制御部138はランダムに1つを選択する。さらに、制御部138は、当該セットにおける選択され



たウェイのキャッシュエントリを対象にリプレースし（ステップS94）、リプレース後に当該キャッシュエントリの使用フラグUを1に、ニューフラグを1に初期化する（ステップS95）。なお、このときバリッドフラグV、ダーティフラグDは、それぞれ1、0に初期化される。

#### 【0061】

このように、リプレース対象は、ニューフラグが0でかつ使用フラグが0のキャッシュエントリを1つ選択することにより決定される。ただし、8つのニューフラグの全てが1である場合には、ニューフラグが1でかつ使用フラグUが0のウェイの中からリプレース対象を1つ選択する。このリプレースアルゴリズムは、従来のLRU方式におけるアクセス順序を示すデータの代わりに1ビットの使用フラグを用いるので、擬似的なLRU方式といえることができる。

#### 【0062】

以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、設定部372を備えることにより、キャッシュメモリ自身が実行中のタスクを判別して、判別したタスクに対応するアクティブウェイデータをウェイ・レジスタ371に設定し、タスク毎にアクティブなウェイの切り替えることができる。その結果、実施の形態1と同様に、タスク切り替えに伴う無駄なリプレースの発生を低減させることができ、ヒット率を向上させることができる。

#### 【0063】

また、本実施の形態におけるキャッシュメモリによれば、従来のLRU方式におけるアクセス順序を示すデータをキャッシュエントリ毎に設ける代わりに、1ビットの使用フラグをキャッシュエントリ毎に設けている。これにより、従来のアクセス順序データを更新する複雑な回路を、使用フラグを更新する簡単なフラグ更新回路（フラグ更新部39）に置き換えることができる。また、リプレース部40において、リプレース対象を、使用フラグが0のキャッシュエントリの1つを選択することにより簡単に決定することができる。このように、本実施の形態におけるキャッシュメモリによれば、ハードウェア規模を大きく低減することができる。しかも、従来のLRUと比較してもほぼ同等のヒット率を得ることができる。

#### 【0064】

さらに、本実施の形態における制御部138は、ニューフラグが1の場合は、当該キャッシュエントリをリプレース対象から除外している。これは、次の理由による。すなわち、使用フラグUは初期値が1であるが他のウェイの使用フラグが順次1になれば、0にリセットされる。つまり、使用フラグUが0のキャッシュエントリであつても一度もアクセスされていない場合がある。こうして使用フラグが0になった場合、リプレース後に一度もアクセスされていないキャッシュエントリが、キャッシュミスの発生により再度リプレース対象に選択されてしまう可能性がある。そのため、ニューフラグNを設けることにより、リプレースされた後に一度もアクセスされていないキャッシュエントリがリプレースされてしまうことを防止することができる。

#### 【0065】

##### <変形例>

なお、本発明のキャッシュメモリは、上記の実施の形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。

(1) 実施の形態1における変形例(1)～(6)を本実施の形態に適用してもよい。

(2) プロセッサ1から実行中のタスクを示すタスク番号（あるいはスレッド番号、プロセス番号等）が出力される場合には、上記判別部100a～100dの代わりに、タスク番号を保持および更新するタスク番号保持部を備える構成としてもよい。この場合セクタ111は、タスク番号に対応するアクティブウェイデータを選択すればよい。

(3) 制御部138は、図11の下段に示したようにセット内の他のウェイの使用フラグUが全部1であれば0にし、ヒットしたウェイ自身の使用フラグUを1に更新するが、この代わりに、ヒットしたウェイ自身の使用フラグも0に更新する構成としてもよい。



(4) ウェイ・レジスタ 371 の内容が更新されたとき、全ての使用フラグをリセットする構成としてもよい。こうすれば、インアクティブなウェイがインアクティブであるがゆえにアクセス順序が古くなっている場合に、インアクティブなウェイがアクティブなウェイに切り替わった直後にリプレースされてしまう可能性を小さくすることができる。

(5) 上記実施形態におけるニューフラグを有しない構成としてもよい。

#### 【0066】

(実施の形態 3)

実施の形態 1、2 では、キャッシュエントリーのリプレース単位がライン (128 バイト) 単位でなされる構成を開示したが、本実施の形態では、リプレース単位がタスク毎にライン単位とサブライン (32 バイト) 単位とで切り替え可能な構成について説明する。

#### 【0067】

図 14 は、本発明の実施の形態 3 におけるキャッシュメモリの構成を示すブロック図である。同図のキャッシュメモリは、図 8 に示した構成と比較して、ウェイ 131a~131h の代わりにウェイ 231a~231h を備える点と、セクタ 233a~233h が追加された点と、制御部 138 の代わりに制御部 238 を備える点とが異なっている。以下、同じ点は説明を省略して異なる点を中心に説明する。

#### 【0068】

ウェイ 231a~231h は、図 8 のウェイ 131a~131h と比べて、キャッシュエントリー内にバリッドフラグとダーティフラグとを 1 ビットずつ保持するのではなく、サブライン毎に保持する点が異なっている。図 15 に、キャッシュエントリーのビット構成を示す。同図のように、同図のように 1 つのキャッシュエントリーは、バリッドフラグ V0~V3、タグ、ラインデータ、使用フラグ U、ニューフラグ N、ダーティフラグ D0~D3 を保持する。使用フラグ U およびニューフラグ N については既に説明したので省略する。ラインデータ (128 バイト) は 4 つのサブライン (32 バイト) からなる。バリッドフラグ V0~V3 は、4 つのサブライン 0~3 に対応し、対応するサブラインが有効か否かを示す。ダーティフラグ D0~D3 は、4 つのサブライン 0~3 に対応し、対応するサブラインに書き込みがあったか否かを示す。バリッドフラグおよびダーティフラグがサブライン単位に設けられているのは、リプレースをサブライン単位でも行うことを可能にするためである。また、ライトバック (又はライトスルー) もサブライン単位で行うことが可能である。

#### 【0069】

セクタ 233a は、ウェイ 231a から、ソースインデックス SI により選択されたセットに対応するバリッドフラグ V0~V3 と、ワードインデックス WI の上位 2 ビットとが入力され、この上位 2 ビットに指定されるサブラインに対応するバリッドフラグを選択する。セクタ 233b~233h についても、ウェイ 231b~231h に対応している点以外同様である。これによりセクタ 233a~233h は、サブライン単位でヒットしたか否かを判定すること可能にしている。

#### 【0070】

制御部 238 は、制御部 138 と比べて、設定部 372 が削除された点と、RS (リプレースサイズ) レジスタ 373 が追加された点とが異なる。

設定部 372 が削除されているのは、ウェイ・レジスタ 371 が実施の形態 1 と同様にタスク切り替えにおいて書き換えられるからである。

RS レジスタ 373 は、ウェイ毎にリプレースサイズを示すリプレースサイズデータを保持する。図 16 に RS レジスタ 373 のビット構成例を示す。同図のように RS レジスタ 373 は、RS0~RS7 からなるリプレースサイズデータを保持する。RS0~RS7 の各ビットは、1 のときリプレースサイズがサブライン (32 バイト) であることを、0 のときリプレースサイズがライン (128 バイト) であることを制御部 238 に指示する。この RS レジスタ 373 は、ウェイ・レジスタ 371 と同様に、プロセッサ 1 から読み書き可能であり、コンテキストの一部としてタスク切り換えにおいて書き換えられる。これにより、リプレースサイズをラインとするとサブラインとするかを、ウェイ毎にかつ

タスク毎に設定することを可能にしている。

#### 【0071】

図17は、制御部238におけるフラグの更新処理を示すフローチャートである。同図は、図12に示したフローチャートと比べて、ダーティフラグをサブライン単位で設定するためのステップS175～S177が追加された点が異なる。すなわち、制御部238は、キャッシュメモリへの書き込みがあったとき（ステップS175）、書き込まれたサブラインを判別し（ステップS176）、判別されたサブラインに対応するダーティフラグを1にセットする（ステップS177）。ステップS175～S177の処理は、例えば、制御部238に1入力4出力のデマルチプレクサをウェイ毎に備えることにより簡単に実現することができる。このデマルチプレクサは、論理“1”が入力され、4つの出力をキャッシュエントリ中のダーティフラグD0～D3に対応させ、ワードインデックスWIの上位2ビットにより出力先を制御するよう構成すればよい。

#### 【0072】

このようにして、制御部238は、サブライン単位に設けられたダーティフラグD0～D3をキャッシュライトに応じて更新する。

図18は、制御部238におけるリプレース処理を示すフローチャートである。同図は、図13に示したフローチャートと比べて、ステップS94の代わりにステップS181～183を有する点と、ステップS95の代わりにステップS95aを有する点とが異なる。

#### 【0073】

制御部238は、RSレジスタ373からステップS93で選択されたウェイに対応するRSフラグを読み出して、リプレースサイズとしてサブラインとラインの何れが指定されているかを判定し（ステップS181）、サブラインと指定されている場合には、当該ウェイのサブラインをリプレースし（ステップS182）、ラインと指定されている場合には、当該ウェイのラインをリプレースする（ステップS182）。さらに、制御部238は、リプレースされたサブラインまたはラインに対応するバリッドフラグおよびダーティフラグを初期化する（ステップS95a）。すなわち、リプレースされたサブラインに対応するバリッドフラグ、ダーティフラグをそれぞれ1、0に設定する。ライン単位でリプレースされた場合には、4つのサブラインに対応する4つのバリッドフラグ、ダーティフラグをそれぞれ1、0に設定する。

#### 【0074】

以上説明してきたように本実施の形態におけるキャッシュメモリによれば、実施の形態1又は2に加えて、リプレース単位をラインとサブラインとでウェイ毎およびタスク毎に設定できるので、タスクの必要とするデータサイズに応じてリプレース単位を切り換えることにより、キャッシュミスをもさらに低減することができる。例えば、タスク1はオーディオデータのデコード/エンコード処理を、タスク2がビデオデータのデコード/エンコード処理を行うものとする。この場合、タスク1ではラインサイズをリプレース単位とし、タスク2ではサブラインをリプレース単位とすることができる。こうすれば、タスク1、2のキャッシュ利用効率を向上させることができる。なぜなら、タスク1は、オーディオデータに対するメモリアクセスをシーケンシャルに行うケースが多く、タスクキャッシュメモリ2は、オーディオデータに対するメモリアクセスをランダムに行うケースが多いからである。

#### 【0075】

##### <変形例>

なお、本発明のキャッシュメモリは、上記の実施の形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。

(1) 実施の形態2における変形例(1)、(3)、(4)、(5)を本実施の形態に適用してもよい。

(2) 上記実施の形態では、サブラインのサイズをラインのサイズの1/4としているが、1/2、1/8、1/16等他のサイズでもよい。その場合、各キャッシュエントリ

は、サブラインと同数のバリッドフラグおよびダーティフラグをそれぞれ保持すればよい。

#### 【0076】

##### (実施の形態4)

実施の形態1～3では、インアクティブなウェイに対して少なくともリブレースが制限される例を説明したが、本実施の形態では、さらに比較器32a～32hのうちインアクティブなウェイに対応する比較器における比較の禁止と、インアクティブなウェイに対応するキャッシュエントリからのタグ出力の禁止をする構成について説明する。

#### 【0077】

そのため本実施の形態におけるキャッシュメモリは図14に示した制御部238内部に比較制御部372を追加した構成となっている。

図19は、比較制御部372およびウェイ231a～231hの要部の構成を示すブロック図である。

同図において、キャッシュアドレスエントリ300a～300hは、キャッシュメモリ中のウェイ0～8に含まれ、それぞれセット数と同数(実施形態では16個)のタグを保持する。キャッシュアドレスエントリ300aは、ウェイ0に含まれ、16個のタグ保持部301～316と、16個のアンド回路321～336とを含む。他のキャッシュアドレスエントリ300b～300hも同様である。

#### 【0078】

アンド回路321は、16個のセット中のセット0に対応し、セットインデックスをデコードするデコーダ30によりセット0が選択信号(set0)と、イネーブル信号E0とのアンドをとる。その結果、アンド回路321は、選択信号set0=1でかつイネーブル信号E0=1のときのみ、タグ保持部301からのタグ出力と、比較器32aとをイネーブルにする。

#### 【0079】

比較器32a～32hは、それぞれイネーブル端子ENを有し、イネーブル端子の入力が1のときに、アドレスレジスタ20中のタグアドレスとキャッシュアドレスエントリ301からのタグとの比較動作を行う。

比較制御部372は、8つのウェイに対応する8つのイネーブル回路381a～381h、回数カウンタ382を備え、比較器32a～32hにおいて最大2回のタグ比較を行わせるよう制御し、1回目のタグ比較では、ウェイ・レジスタ371に示されたアクティブなウェイに対応する比較器をイネーブルにし、かつインアクティブウェイに対応する比較器をディスエーブルすることによって、アクティブウェイのタグを比較対象とし、さらに、1回目のタグ比較においてミスヒットと判定された場合に、アクティブウェイに対応する比較器をディスエーブルし、アクティブウェイに対応する比較器をイネーブルにすることによって、インアクティブウェイのタグを比較対象とし2回目のタグ比較を行わせるよう構成している。また、各比較器のイネーブル/ディスエーブルの制御と同時にウェイ231a～231hからのタグ出力もイネーブル/ディスエーブルを制御している。これによりディスエーブルされた比較器およびタグ出力による消費電力の低減を図っている。

#### 【0080】

イネーブル回路381a～381hは、ウェイ231a～231hに対応し、ウェイ・レジスタ371に保持されるアクティブウェイデータに従って、比較器32a～32hのうち、1回目のタグ比較ではアクティブなウェイに対応する比較器のみをイネーブルにし、2回目のタグ比較ではインアクティブなウェイに対応する比較器のみをイネーブルにする。

すなわち、イネーブル回路381a～381hは、イネーブル信号E0～E7を生成し、このイネーブル信号E0～E7により、キャッシュメモリ中のウェイ0～8に対応する8つのキャッシュアドレスエントリからのタグ出力と、8つの比較器32a～32hをイネーブル/ディスエーブルする。例えば、イネーブル回路381aは排他的論理和回路により、ウェイ0がアクティブか否かを示すW0ビットと、回数カウンタ382のカウント

値に従ってイネーブル信号E 0を生成する。

【0081】

回数カウンタ382は、比較の回数をカウントするカウンタであり、0（1回目）、1（2回目）とカウントアップする。ただし、1回目がヒットした場合にはカウントアップしない。

図20は、回数カウンタ382のカウント値とアクティブウェイデータW<sub>n</sub>（n=0～7）を入力として、イネーブル信号E<sub>n</sub>を出力とするイネーブル回路の制御論理を示す真理値表を示す。同図において、例えば、ウェイ0～2がアクティブでウェイ3～7がインアクティブである場合（W0=1）、1回目の比較では、イネーブル信号E0～E2が1（イネーブル）でイネーブル信号E3～E7がディスエーブルになる。この1回目の比較でヒットした場合には、回数カウンタがカウントアップしないので2回目の比較はなされない。1回目の比較でミスヒットした場合には、回数カウンタがカウントアップするので2回目の比較がなされる。この場合、2回目の比較では、イネーブル信号E0～E2が0（ディスエーブル）でイネーブル信号E3～E7が1（イネーブル）になる。

【0082】

以上の構成により、1回目のタグ比較では、インアクティブなウェイを比較対照としないので比較器における消費電力およびタグ出力による消費電力を低減することができる。さらに、1回目のタグ比較においてミスヒットした場合には、インアクティブなウェイのみを比較対象として2回目のタグ比較を行うので、全てのウェイのキャッシュデータを有効に活用することができる。

【0083】

なお、2回目の比較において全ての比較器をイネーブルにしてもよい。この場合、1回目でのヒットした場合に2回目の比較が行われないので、消費電力を低減することができる。つまり、タスクごとにアクティブウェイが割り当てられるので1回目の比較において高いヒット率を得られると考えられる。

また、2回目の比較を行わない構成としてもよい。この1回目の比較において高いヒット率を得られると考えられるので、消費電力を低減する効果がある。

【産業上の利用可能性】

【0084】

本発明は、メモリアクセスを高速化するためのキャッシュメモリおよびその制御方法に適しており、例えば、オンチップキャッシュメモリ、オフチップキャッシュメモリ、データキャッシュメモリ、命令キャッシュメモリ等に適している。

【図面の簡単な説明】

【0085】

【図1】本発明の実施の形態1におけるプロセッサ、キャッシュメモリ、メモリを含むシステムの概略構成を示すブロック図である。

【図2】キャッシュメモリの構成例を示すブロック図である。

【図3】ウェイ・レジスタのビット構成を示す図である。

【図4】ウェイ・レジスタとウェイとの対応関係を示す説明図である。

【図5】制御部におけるリプレース処理を示すフローチャートである。

【図6】タスク切り替え処理を示すフローチャートである。

【図7】タスク領域のコンテキストに含まれるタスク毎のウェイデータを示す図である。

【図8】本発明の実施の形態2におけるキャッシュメモリの構成を示すブロック図である。

【図9】キャッシュエントリーのビット構成を示す図である。

【図10】設定部の構成を示すブロック図である。

【図11】フラグの更新例を示す説明図である。

【図12】フラグ更新処理フローを示す図である。

【図13】リプレース処理フローを示す図である。

【図 14】本発明の実施の形態 3 におけるキャッシュメモリの構成を示すブロック図である。

【図 15】キャッシュエントリーのビット構成を示す図である。

【図 16】リプレースサイズレジスタのビット構成を示す図である。

【図 17】12 ベース フラグの更新例を示す説明図である。

【図 18】ベース リプレース処理を示すフローチャートである。

【図 19】本発明の実施の形態 4 における比較制御部および各ウェイの要部の構成を示すブロック図である。

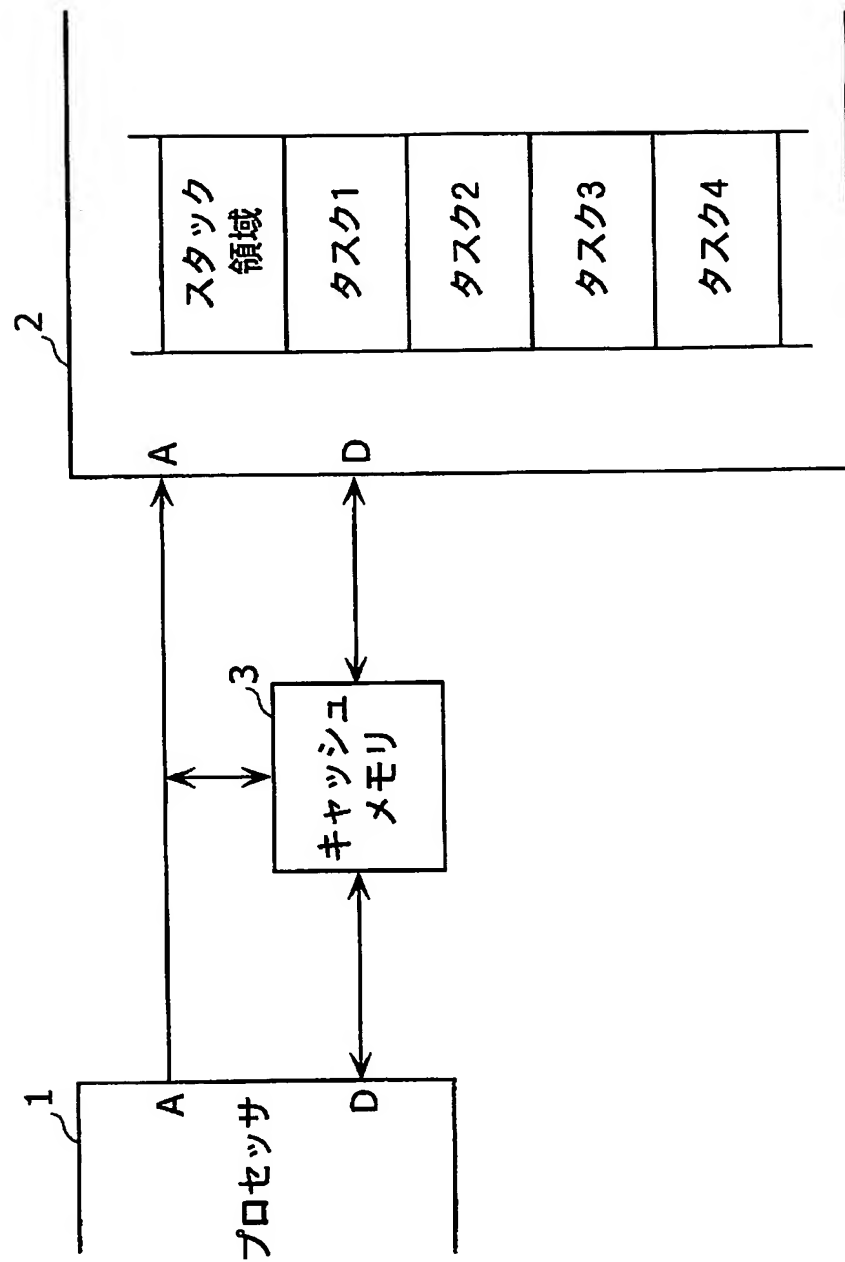
【図 20】イネーブル回路の制御論理を示す真理値表を示す。

【符号の説明】

【0086】

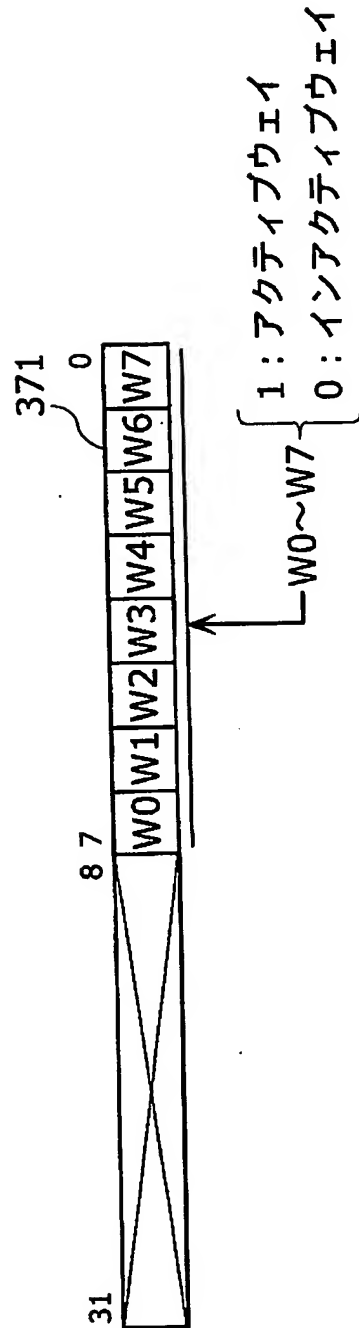
- 1 プロセッサ
- 2 メモリ
- 3 キャッシュメモリ
- 20 アドレスレジスタ
- 30 デコーダ
- 31a~31h ウェイ
- 32a~32h 比較器
- 33a~33h アンド回路
- 34 オア回路
- 35、36 セレクタ
- 37 デマルチプレクサ
- 38 制御部
- 39 フラグ更新部
- 40 リプレース部
- 80~87 アンド回路
- 88~92 オア回路
- 93~96 セレクタ

【書類名】 図面  
【図 1】



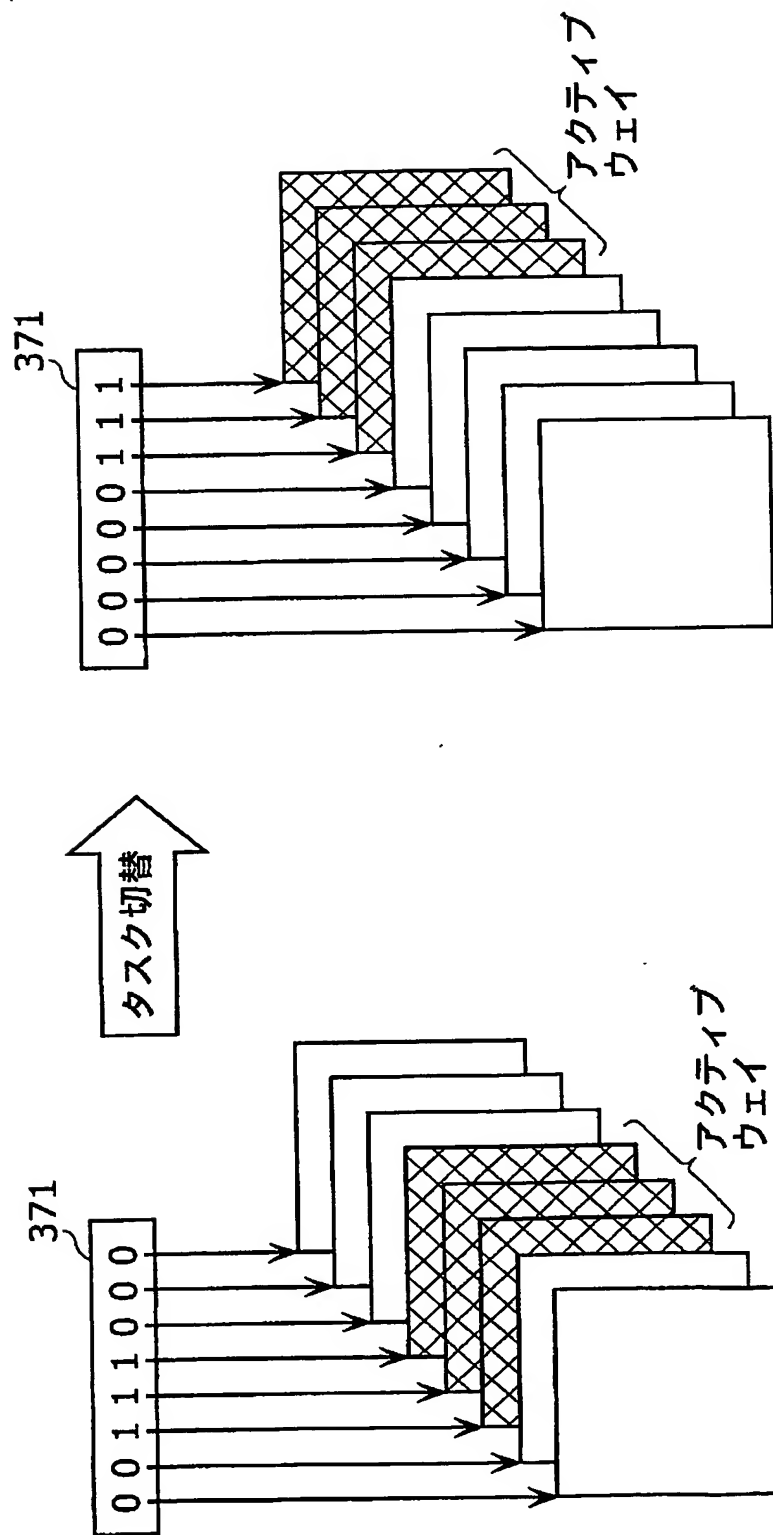


【図 3】

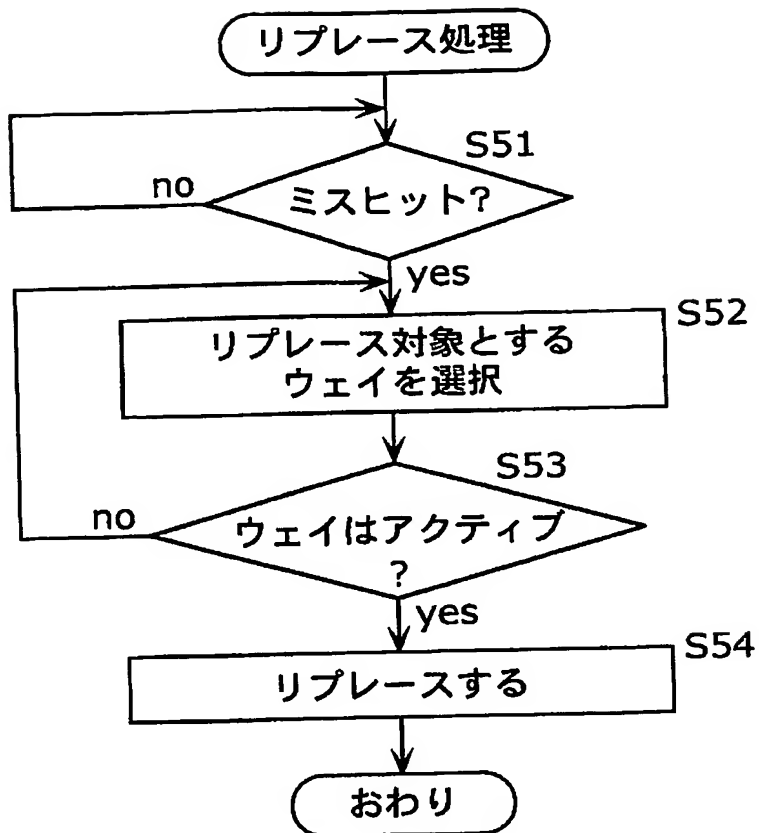




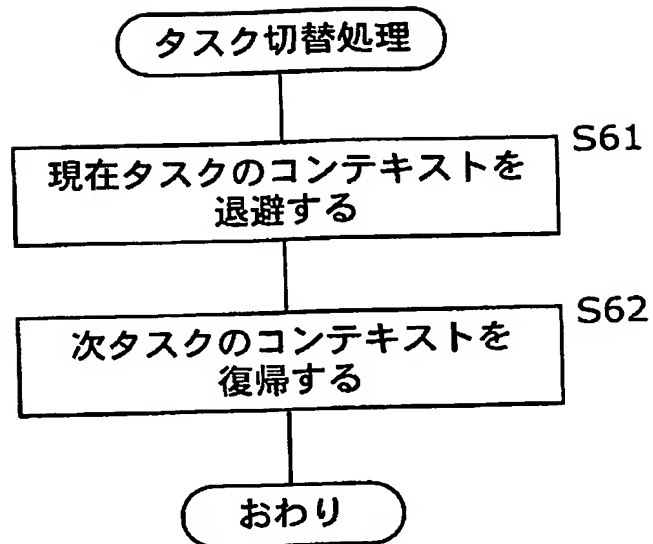
【図 4】



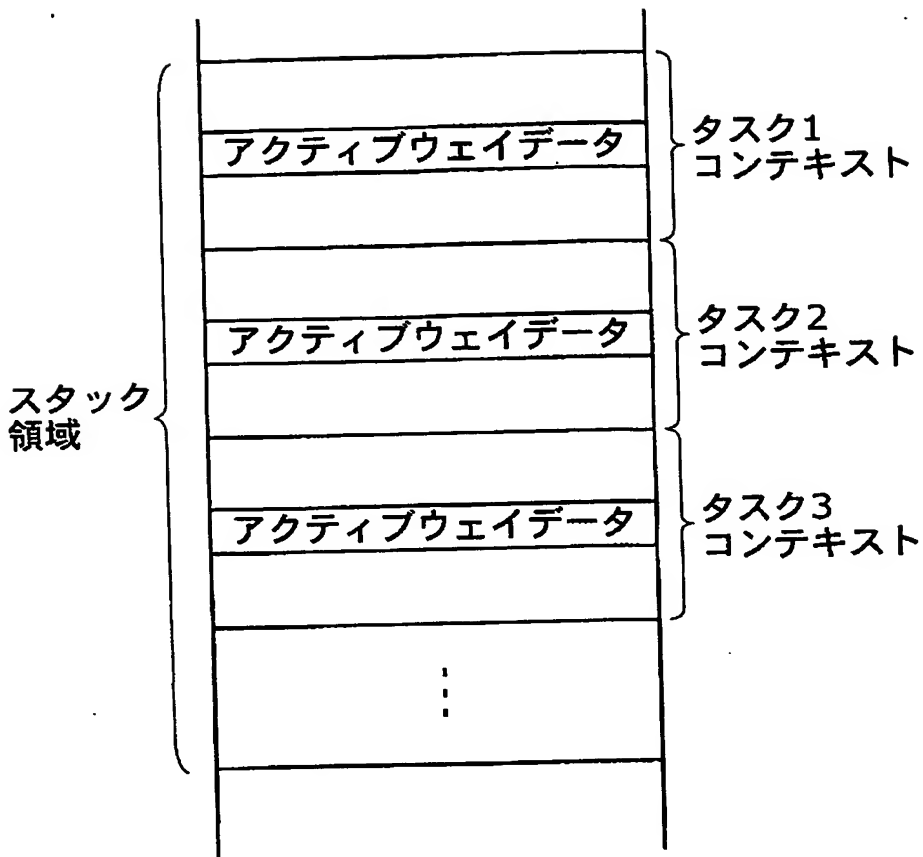
【図 5】



【図 6】

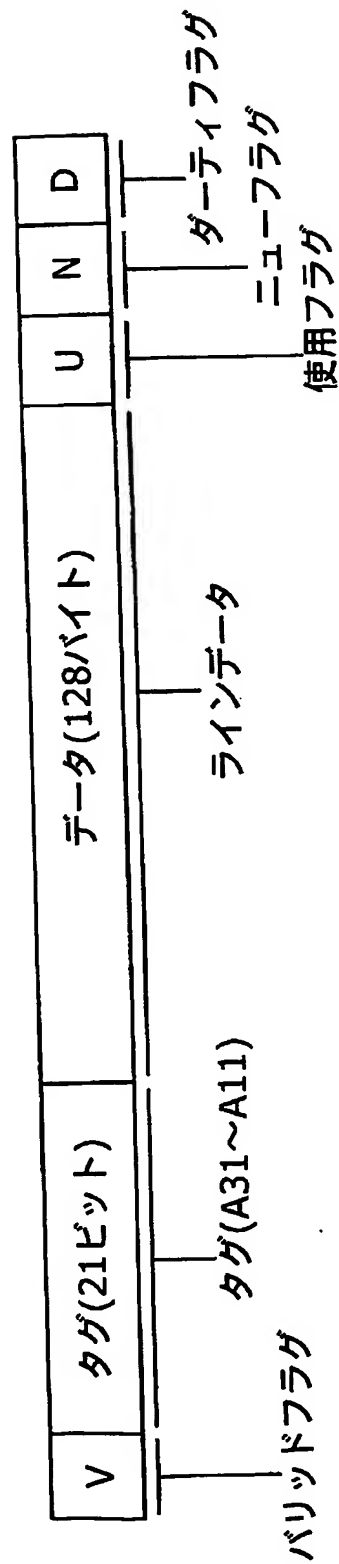


【図 7】

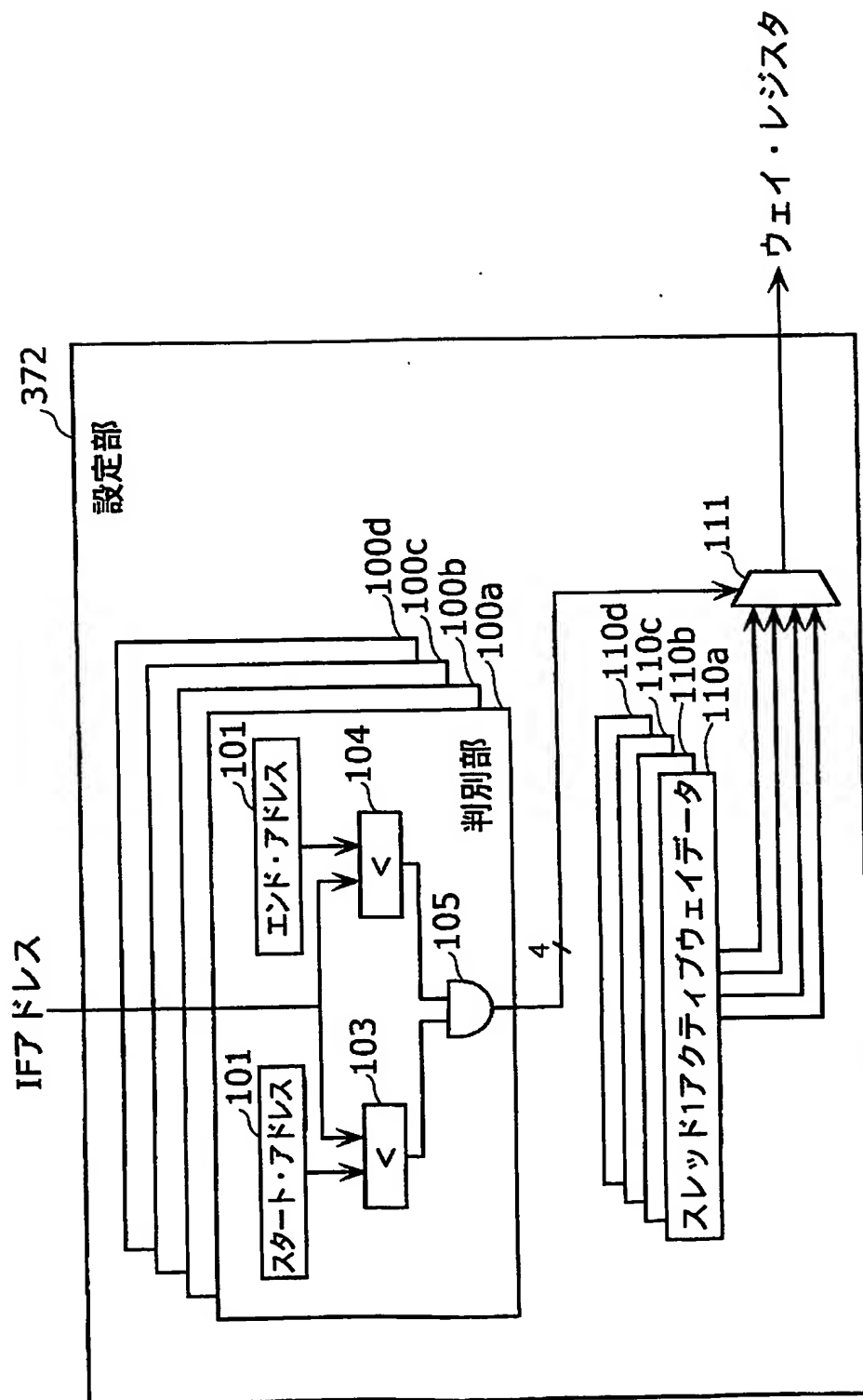




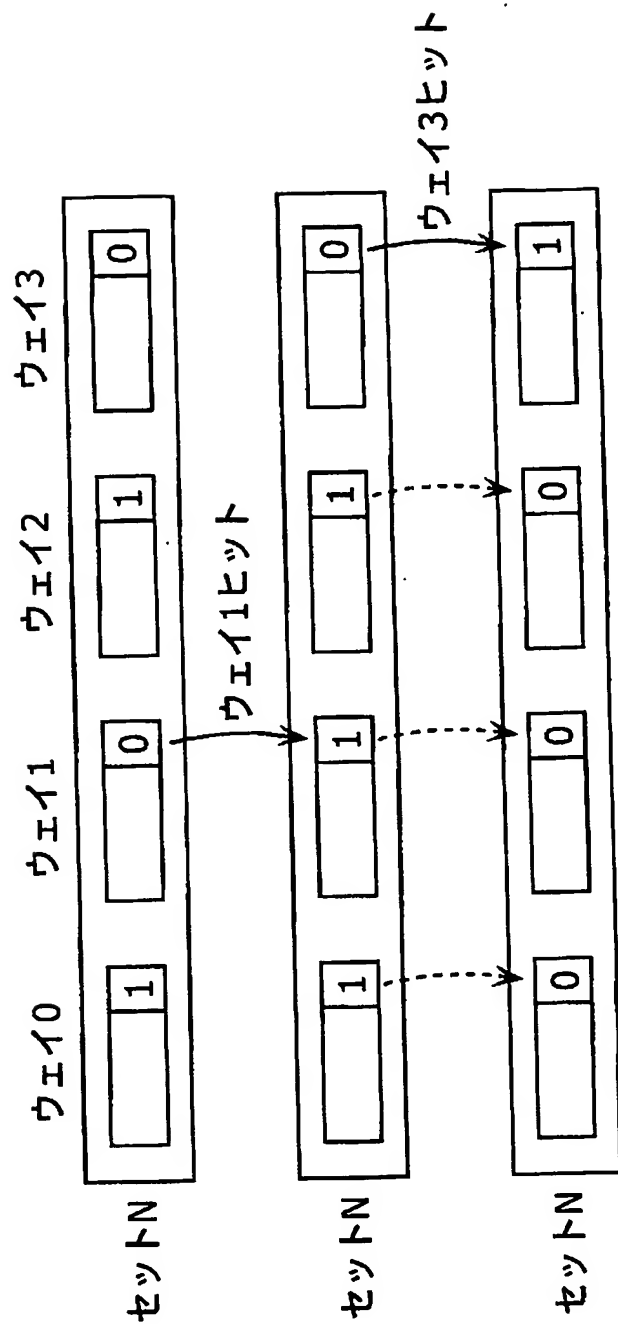
【図 9】



【図 10】

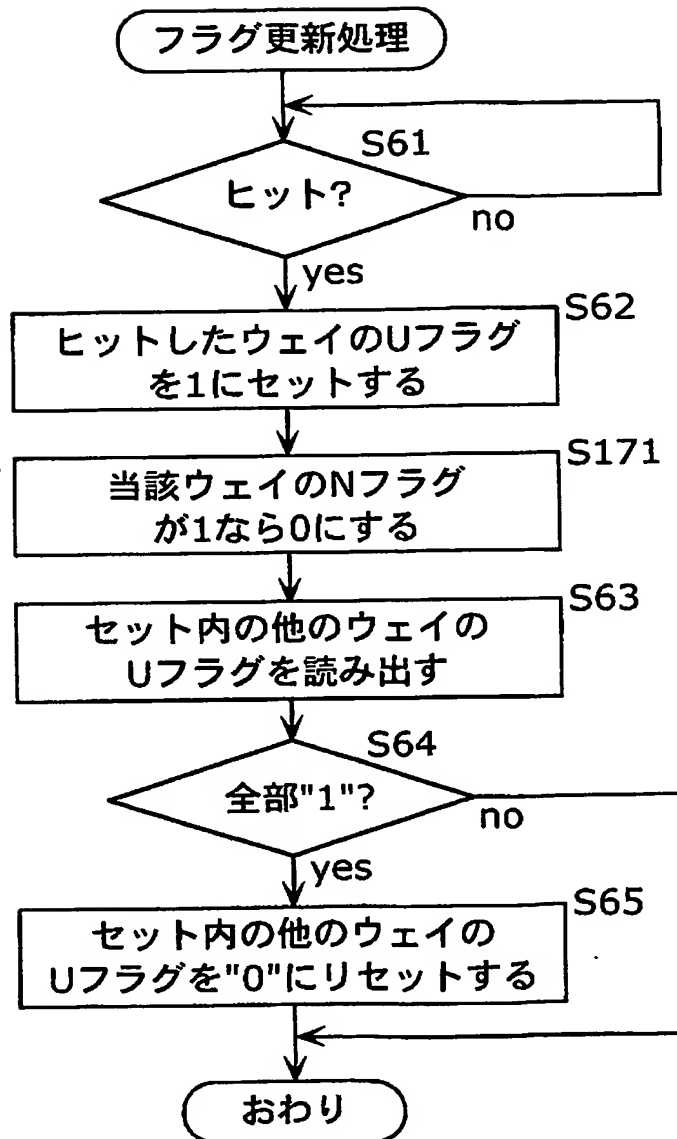


【図11】

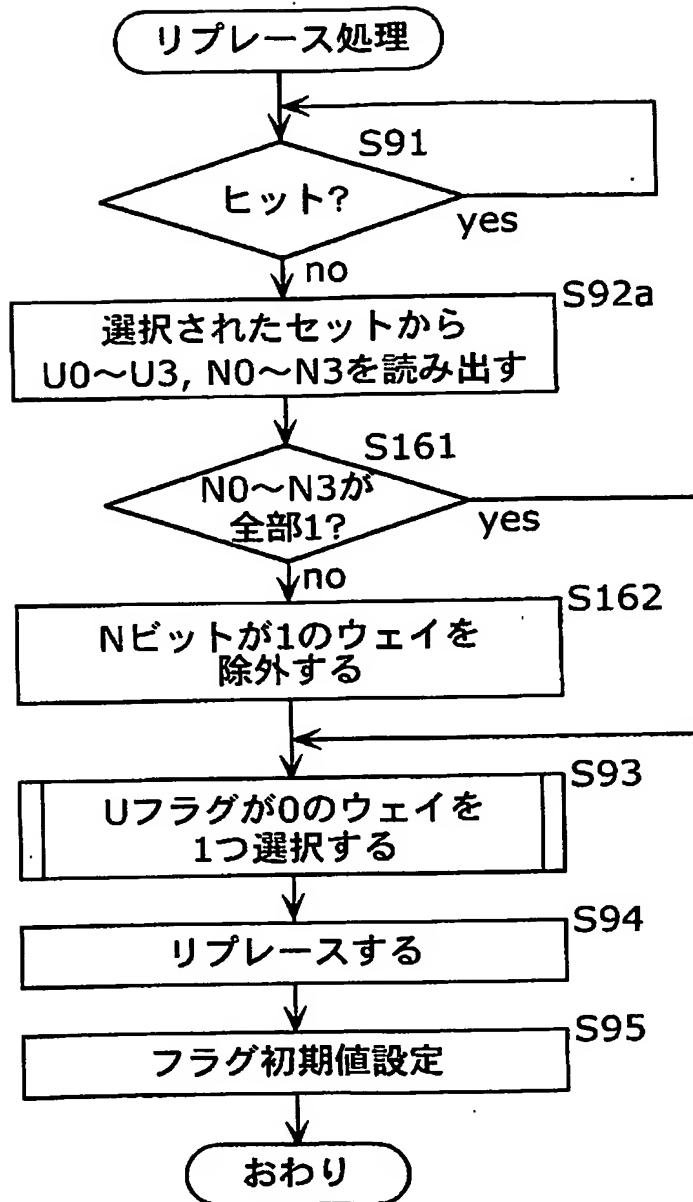




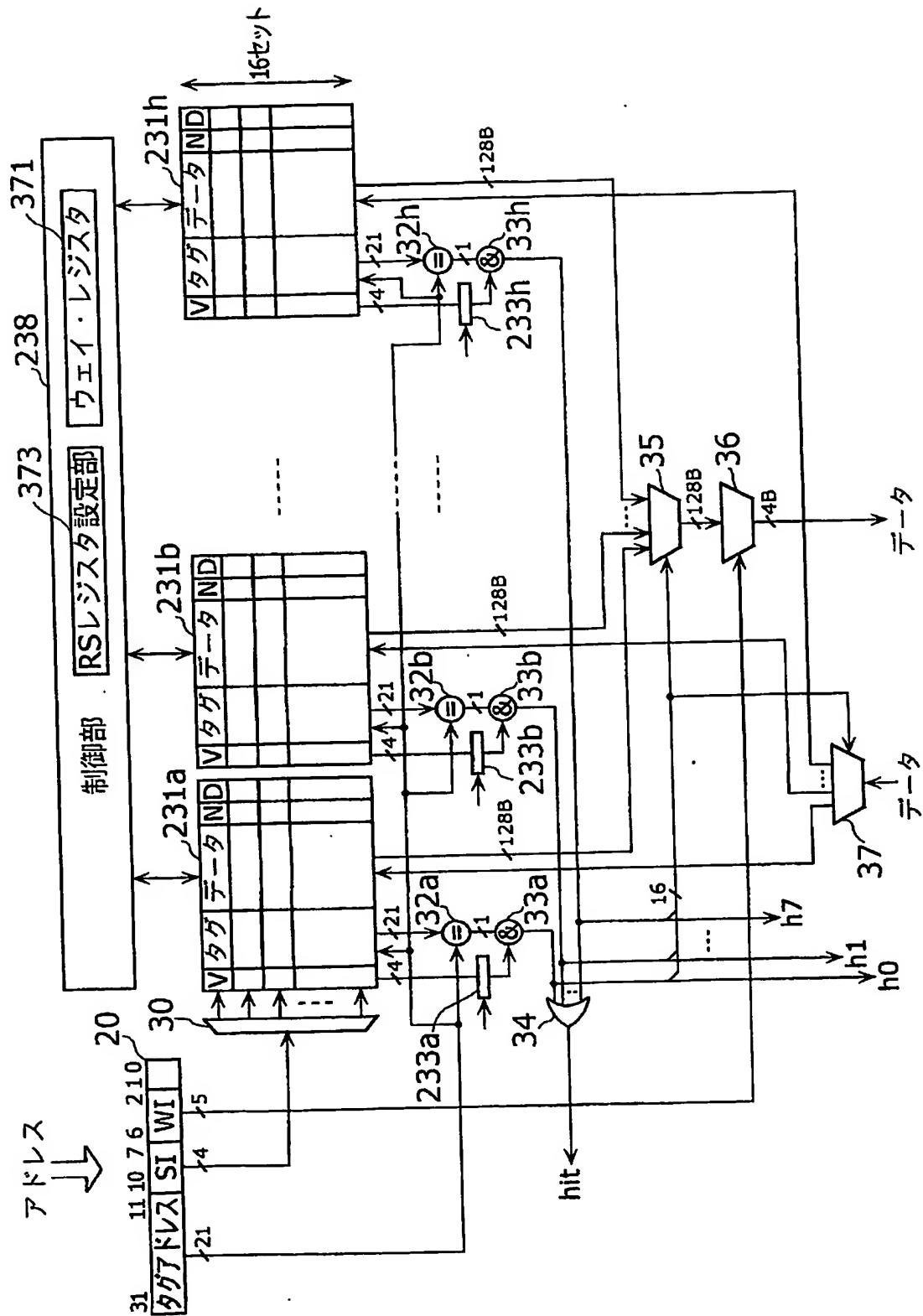
【図 12】



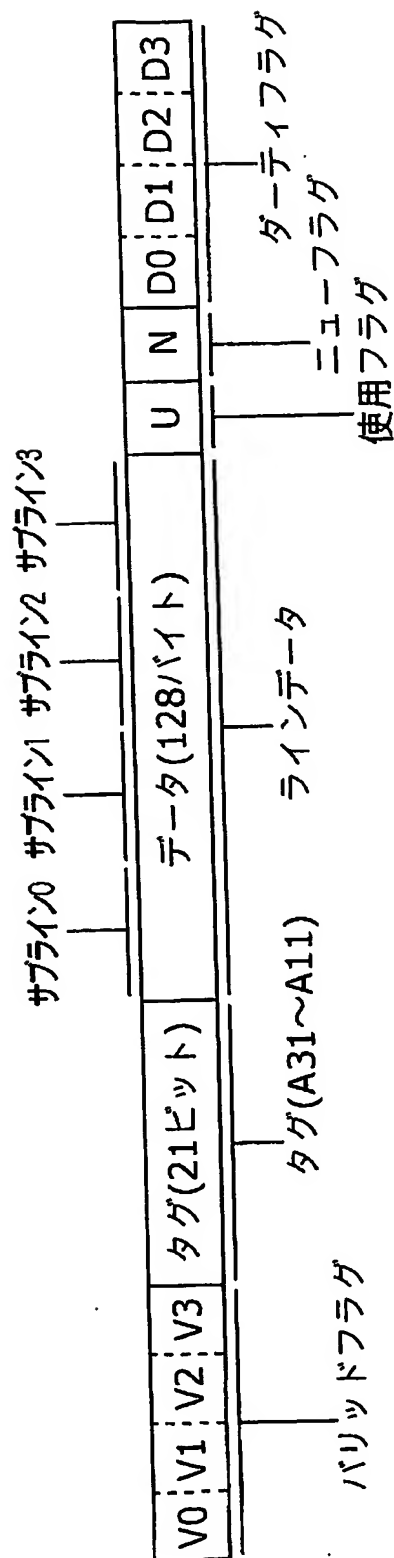
【図 13】



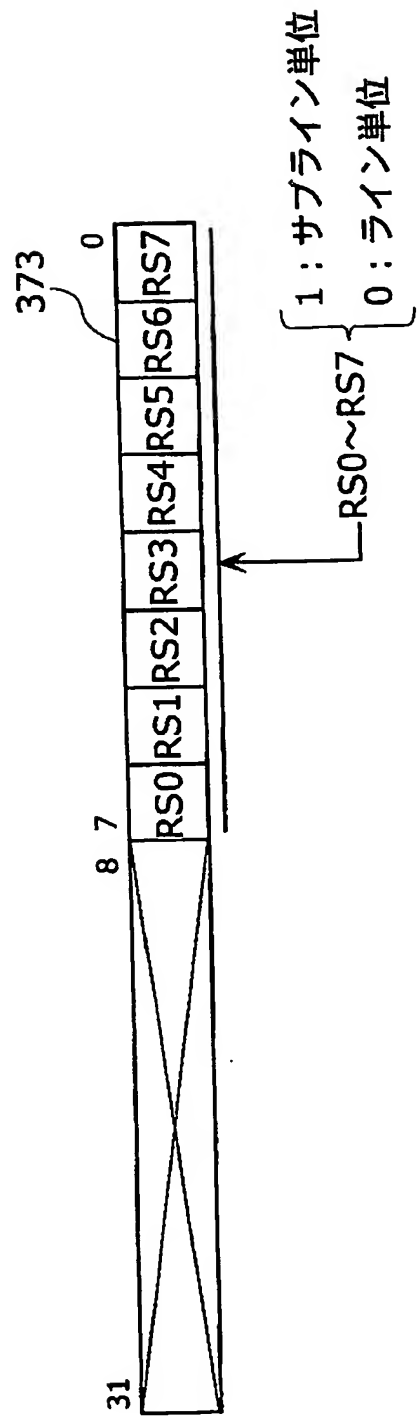
【図 14】



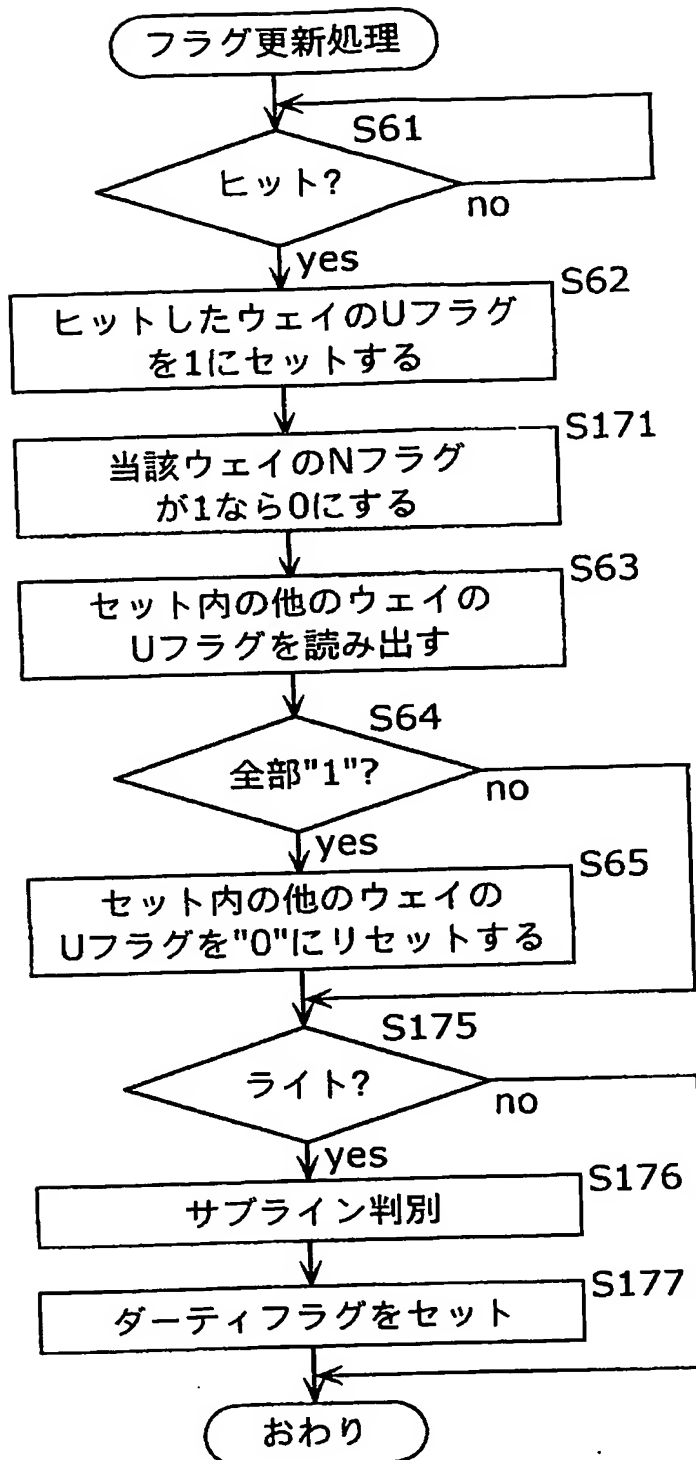
【図 15】



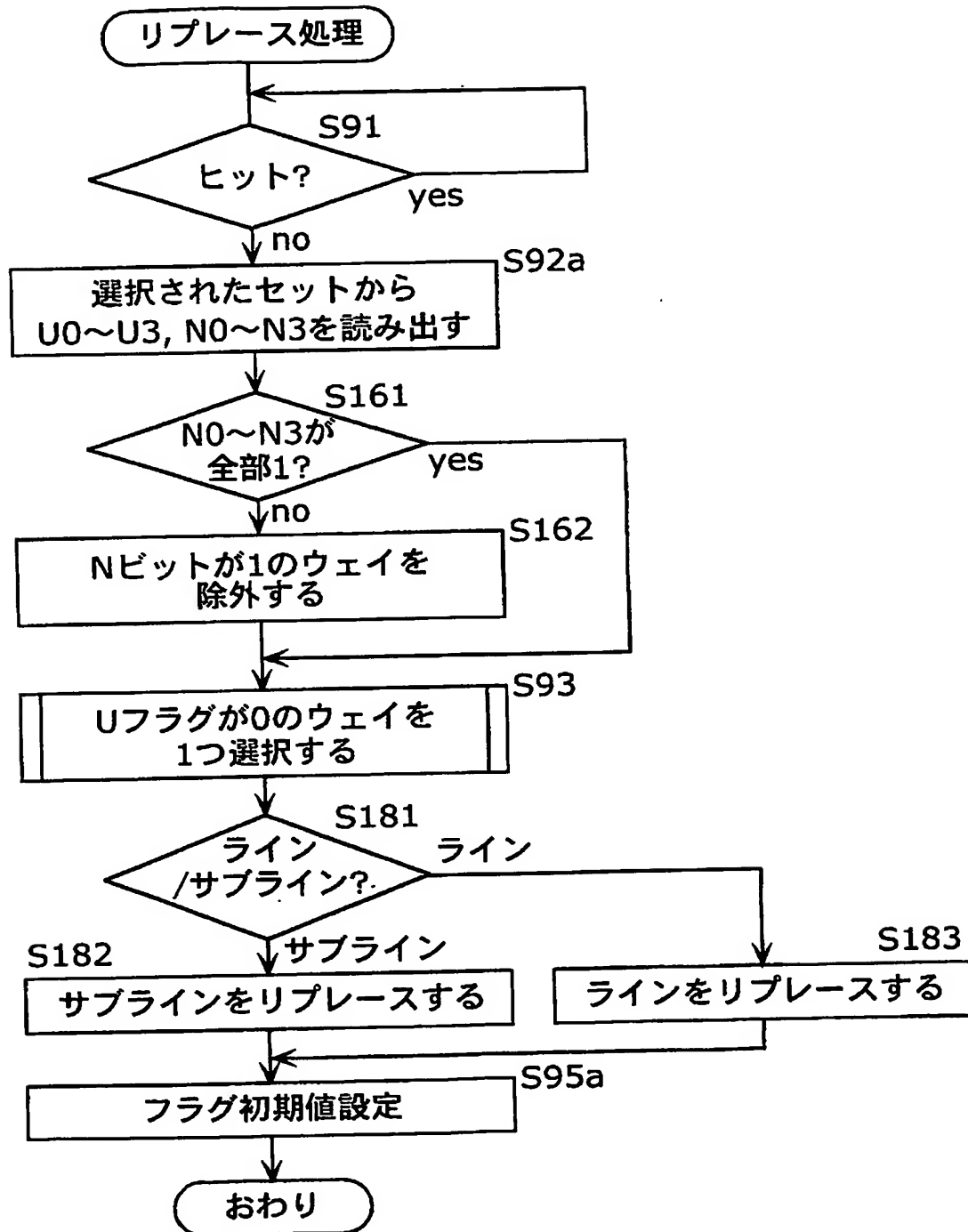
【図 16】



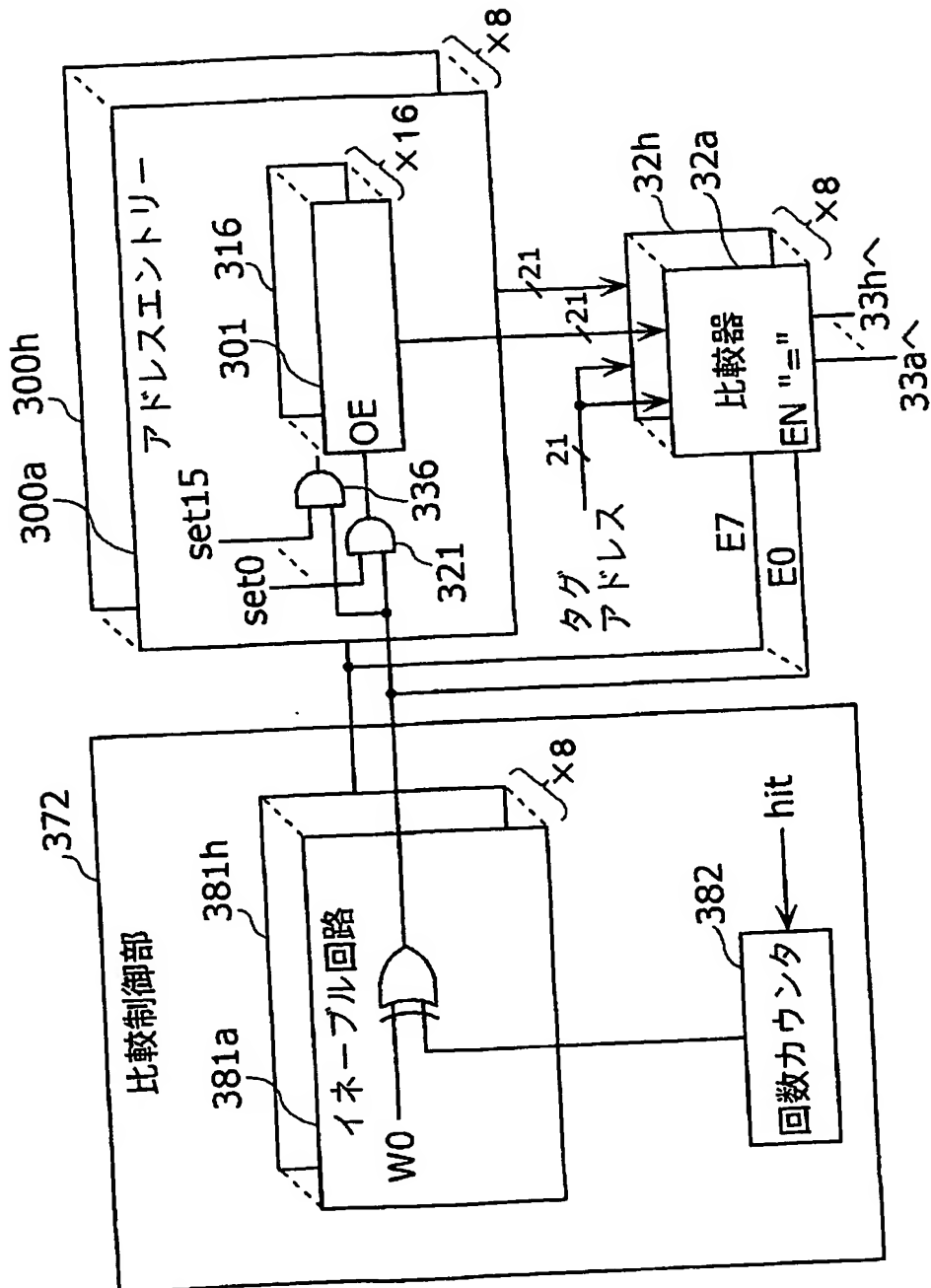
【図 17】



【図18】



【図 19】





【図 2 0】

入力		出力
Wn(n=0~7)	カウント値	En
1(アクティブ)	0(1回目)	1(イネーブル)
1(アクティブ)	1(2回目)	0(ディスエーブル)
0(インアクティブ)	0(1回目)	0(ディスエーブル)
0(インアクティブ)	1(2回目)	1(イネーブル)

## 【書類名】 要約書

## 【要約】

【課題】 タスク切り替え等によるキャッシュメモリのヒット率の低下を防止し、タスクの実質的な処理時間を容易に確保するキャッシュメモリを提供する。

【解決手段】 N-ウェイ・セット・アソシエイティブ方式のキャッシュメモリであって、N個のウェイ毎にアクティブとすべきかインアクティブとすべきかを示すアクティブウェイデータを保持するウェイレジスタ371を有し、タスク切り替えに際してウェイレジスタの内容を更新し、インアクティブなウェイに対してリプレースを制限する。

## 【選択図】 図1

認定・付加情報

特許出願の番号	特願 2003-382570
受付番号	50301871206
書類名	特許願
担当官	第七担当上席 0096
作成日	平成15年11月13日

<認定情報・付加情報>

【提出日】 平成15年11月12日

特願 2003-382570

出願人履歴情報

識別番号

[000005821]

1. 変更年月日

1990年 8月28日

[変更理由]

新規登録

住 所

大阪府門真市大字門真1006番地

氏 名

松下電器産業株式会社